WONS 2006

The Third Annual Conference on Wireless On demand Network Systems and Services

January 18th, 2006 - January 20th, 2006
Les Ménuires, France

http://ares.insa-lyon.fr/wons2006/

Photo Credits : Catherine and Jean-Marc Roussel
WONS Logo courtesy of Mauro Brunato

With the financial support of:

INRIA RHÔNE-ALPES

INSA LYON

ALCATEL

Sponsored by:

ifip tc 6

Preface

Dear Workshop Attendee:

We are proud to present to you the Proceedings of the Third Edition of the Annual Conference on Wireless On-demand Network Systems and Services (WONS 2006). After Madonna di Campiglio (Italy) and St Moritz (Switzerland), the conference takes place this year in Les Ménuires, set in the beautiful French Alps, from January, 18th to January, 20th, 2006. The conference was organized by INRIA and sponsored by IFIP, INSA de Lyon and Alcatel. We wish to thank all the sponsors for their organizational support and generous contributions.

The WONS 2006 event consolidates the success of the previous editions of the conference series. It firmly establishes WONS as the ideal forum for researchers from academia, research laboratories and industry from all over the world to come and share their ideas, views, results and experiences in wireless on-demand networks, systems and services.

This year, 56 papers were submitted. From the Open Call submissions we accepted 16 papers as full papers (up to 12 pages) and 8 papers as short papers (up to 6 pages). All the accepted papers will be presented orally in the Workshop sessions. More precisely, the selected papers have been organized in 7 session: Channel access and scheduling, Energy-aware Protocols, QoS in Mobile Ad-Hoc networks, Multihop Performance Issues, Wireless Internet, Applications and finally Security Issues. The papers (and authors) come from all parts of the world, confirming the international stature of this Workshop. The majority of the contributions are from Europe (France, Germany, Greece, Italy, Netherlands, Norway, Switzerland, UK). However, a significant number is from Australia, Brazil, Canada, Iran, Korea and USA. The proceedings also include two invited papers. We take this opportunity to thank all the authors who submitted their papers to WONS 2006. You helped make this event again a success!

The review process was thorough. For each paper, we obtained at least two and typically three reviews. This could only be realized through the hard work of an outstanding Technical Program Committee composed of 39 members. We thank all the TPC members for their invaluable help with the paper reviews.

We wish to express our deep gratitude to Serge Fdida for accepting our invitation to deliver the keynote message to the conference, entitled Networking in Autonomic Communications. We also wish to thank Renato Lo Cigno for organizing a panel on the still very timely theme: Is There a Killer Application for On-Demand Networking? We hope the answer will be YES, of course, and we thank all the panelists in advance for agreeing to offer their vision and opinions on this controversial topic. We sincerely thank the authors of our two invited papers: Pan Hui (University of Cambridge), James Scott (Intel Research), Jon Crowcroft (University of Cambridge), and Christophe Diot (Thomson Research), will present Haggle: a networking architecture designed around mobile users, and Longbi Lin (Purdue University), Ness B. Shroff (Purdue University), and R. Srikant (UIUC) will present Energy efficient Routing in Sensor Networks: A Large Systems Approach.

Finally, the success of the Workshop is due in no little part to the hard work of many colleagues. In particular, we wish to thank Danièle Herzog for handling the organizational aspects and Claude Chaudet for taking on all sorts of charges (including emergencies) as a true jack-of-all-trades!

We wish you a profitable participation to WONS 2006 and a productive interaction with your peers, either during the sessions or on the alpine slopes!

Mario Gerla and Isabelle Guérin Lassous, General Chairs
Carla-Fabiana Chiasserini and Edward W. Knightly, Conference Program Chairs

General Chairs

Mario Gerla	UCLA, Los Angeles, CA, USA
Isabelle Guérin Lassous	INRIA/CITI, Lyon, France

Program Committee Chairs

Carla-Fabiana Chiasserini	Politecnico di Torino, Torino, Italy
Edward W. Knightly	Rice University, Houston, TX, USA

Steering Committee

Roberto Battiti (chair)	University of Trento, Trento, Italy
Imrich Chlamtac	CREATE-NET, Trento, Italy
Mario Gerla	UCLA, Los Angeles, CA, USA
Enrico Gregori	CNR, Pisa, Italy
Ioannis Stavrakakis	University of Athens, Athens, Greece

Organization Committee

Claude Chaudet	ENST, Paris, France
Daniele Herzog	INRIA, Grenoble, France

Technical Program Committee

Arup Acharya	IBM Research
Eitan Altman	INRIA
Victor Bahl	Microsoft Research
Chadi Barakat	INRIA
Michel Barbeau	Carleton University
Dominique Barthel	France Telecom R&D
Roberto Battiti	University of Trento
Christian Bettstetter	DoCoMo Euro-Labs
Giuseppe Bianchi	University of Roma Tor Vergata
Torsten Braun	University of Berne
Levente Buttyan	Budapest University of Technology and Economics
Antonio Capone	Politecnico di Milano
Sunghyun Choi	Seoul National University
Marco Conti	CNR-IIT
Jon Crowcroft	University of Cambridge
Francesca Cuomo	University of Roma "La Sapienza"
Christophe Diot	Intel Research
Andrzej Duda	LSR-IMAG Laboratory
Laura Feeney	Swedish Institute of Computer Science
Luca Gambardella	IDSIA
Silvia Giordano	SUPSI of Lugano
Enrico Gregori	CNR-IIT
Paul Havinga	University of Twente
Thomas Hou	Virginia Tech
Holger Karl	University of Paderborn
Xiang-Yang Li	Illinois Institute of Technology
Renato Lo Cigno	Università di Trento
Martin Mauve	Heinrich-Heine University

Michela Meo	Politecnico di Torino
Jelena Misic	University of Manitoba
Giacomo Morabito	University of Catania
Amiya Nayak SITE	University of Ottawa
Gian Paolo Rossi	Università di Milano
Bahareh Sadeghi	Intel Corp.
Mariagiovanna Sami	Politecnico di Milano
Puneet Sharma	Hewlett Packard Labs
David Simplot-Ryl	University of Lille
Ioannis Stavrakakis	University of Athens
Ivan Stojmenovic	University of Ottawa
Violet Syrotiuk	Arizona State University
Csaba Szabó	Budapest University of Technology and Economics
Stavros Toumpis	Forschungszentrum Telekommunikation Wien
Ljiljana Trajkovic	Simon Fraser University
Phuoc Tran-Gia	University of Wuerzburg
Menzo Wentink	Conexant
Prudence Wong	University of Liverpool
Boon Sain Yeo	Institute for Infocomm Research
Martina Zitterbart	University of Karlsruhe
Michele Zorzi	Università degli Studi di Padova

Table of Contents

Session 1: Multihop Performance Issues

Session 2: Applications

Invited paper

Session 3: Wireless Internet

Session 4: Channel Access and Scheduling

Session 5: Security Issues

Invited paper

Session 6: Energy-aware protocols

Session 7: QoS in Mobile Ad Hoc Networks

Towards End-to-End QoS in Ad Hoc Networks

OSMAN SALEM and ABDELMALEK BENZEKRI
Institut de Recherche en Informatique de Toulouse,
Université Paul Sabatier,
118 Route de Narbonne - 31062 Toulouse Cedex 04 - France
Email:{benzekri, osman}@irit.fr

***Abstract*— In this paper, we address the problem of supporting adaptive QoS resource management in mobile ad hoc networks, by proposing an efficient model for providing proportional end-to-end QoS between classes. The effectiveness of our proposed solution in meeting desired QoS differentiation at a specific node and from end-to-end are assessed by simulation using a queueing network model implemented in QNAP. The experiments results show that the proposed solution provides consistent proportional differentiation for any service class and validates our claim even under bursty traffic and fading channel conditions.**

I. INTRODUCTION

A mobile Ad Hoc network (MANET[1]) is a collection of autonomous mobile hosts, where each one is equipped with a wireless card that makes it able to communicate with any other host, directly if this last is in the same receiving zone, or indirectly through intermediate hosts that forward packets towards the required destination. Therefore, each host acts as a router when cooperating to forward packets for others, as well as a communication end point.

With the evolution of wireless communications and the emergence of diversified multimedia technologies, quality of service in ad hoc networks became an area of great interest. Besides existing problems for QoS in IP networks, the characteristics of MANETs impose new constraints due to the dynamic behavior of each host and the variation of limited available resources.

A lot of researches has been investigated in routing area [1], [2], [3], [4], [5], [6], [7], and today routing protocols are considered mature enough to face energy constraints and frequently changing network topology caused by mobility (e.g., DSR [2], [6], AODV [3], [6], etc.). Many QoS aware routing protocols that claim to provide a partial (or complete) solution to QoS routing problems have appeared consequently, e.g. QoS-AODV [4], MP-DSR [5], ASAP [8], CEDAR [9].

In the current days, the Integrated services (IntServ) [10] and the differentiated services (DiffServ) [11] are the two principal architectures proposed to provide QoS in wired networks. While the IntServ approach achieves end-to-end services guarantees through per-flow resource reservations, the DiffServ focuses on traffic aggregates and provides more scalable architecture. Contrarily to IntServ, DiffServ does not require any per-flow admission control or signaling, and routers do not maintain a per-flow state information. Routers in DiffServ domain, need only to implement a priority scheduling and buffering mechanism in order to serve packets according to specified fields in their headers.

The migration of these architectures to MANETs is proved to be inconsistent with the characteristics of these networks [12], [13]. Many researches have been based on these concepts and the mitigation of their impediments to make them suitable with the characteristics of MANETs, like INSIGNA [14], [15], FQMM [16], and SWAN [17]. Most of these strategies rely on admission control, priority based resource allocation and scheduling. They only ensure that a newly added flow achieves its desired QoS but can not prevent the degradation of existing flows due to the contention with newly admitted flows and link breaking.

Taking into account that the bandwidth fluctuations and routes change over time, existing QoS mechanisms that require explicit resource reservation and absolute QoS guarantee are difficult to realize in ad hoc networks. The mobility and link breaking make useless the network resources reservation to provide hard guarantee when the resources do not exist. Clearly, there are a number of reasons to beleive that per-hop differentiation technique is more appropriate for ad hoc networks than resources reservation. Without loss of generality, DiffServ is addressed to the network core by aggregating flows in a set of classes. Thus allows per-hop differentiated services for the aggregated flows in the core of the network without requiring any resources allocation. However, DiffServ architecture largely depend on available resources and it does not define any scheme for taking corrective actions when congestion occurs. This is why a static DiffServ model is not suitable for ad hoc networks, and it is imperative to use some kind of feedback as a measure of the conditions of the network to dynamically regulate the class of traffic in the network with respect to the perceived and required QoS.

We turn our attention to provide an adaptive approach with a soft guarantee (small time scale violation) rather than absolute one (strict guarantee). Adaptive services are very attractive in ad hoc networks, because networks resources are relatively scarces and widely variables, where resources fluctuations are mostly caused by mobility, energy constraints and channel fading. In contrast to traditional techniques (IntServ and DiffServ), our model adjusts the spacing of QoS perceived by each class proportionally and independently of network load.

Our approach to provide proportional end-to-end QoS in ad hoc is based on the idea of fighting QoS degradation due to mobility, which will change the topology and will produce bandwidth fluctuation due to load redistribution when re-routing existing traffic, after link break caused by node

movement outside the radio range of its vicinity. Therefore, the relayed packets by this node of an active communications flow towards a destination will be inevitably lost. To recover communication, source node initiates end-to-end alternative route discovery with reactive routing protocol, and flows will travel through the newly discovered route if there is one, causing load redistribution and changing the QoS perceived by existing traffic profile in the newly traversed route due to additional amount of traffic.

The Proportional Differentiated Service (PDS) [18], [19] model, which was proposed for IP networks, classifies flows into N classes where class i gets better proportional performance than class $i-1$. PDS aims to achieve better performance for high priority class relatively to low priority class within fixed pre-specified quality spacing. This proportionality is achieved through the use of a scheduling mechanism able to provide the pre-specified spacing between classes. The most important idea of PDS is that even the actual quality of each class will vary with network load, the spacing ratio between classes will remain constant.

A proportional approach outperforms the strict prioritization scheme, where higher priority classes are serviced before lower ones. This means if the high priority classes are persistently backlogged in corresponding queues, the low priority will starve for bandwidth with strict priority scheduling. Moreover, strict priority schemes do not provide a tuning mechanism for adjusting the quality spacing among classes, and the QoS perceived by a class depends only on the load distribution. Another advantage for using proportional differentiated service rather than strict one is that bandwidth degradation is sensed firstly by the low priority flows, whenever it is desirable to distribute the bandwidth degradation across different classes in a proportional fair manner. Furthermore, while some research in ad hoc try to provide a static fairness in resource network distribution, obsessive fairness is neither reasonable nor desirable in this kind of networks, where some applications expect better services than others.

Usually, QoS parameters are specified in term of maximum end-to-end delay, maximum loss rate and minimum throughput. The differentiation between classes in a static manner will not to be able to respond to these end-to-end requirements with resources fluctuations. To resolve this problem, we will use a priority adaptor mechanism (or dynamic class selection mechanism proposed in [20]) to dynamically adjust the priority of each flow according to the perceived/required QoS.

In this paper, we address the problem of providing a proportional differentiation service that supports a wide range variation of the network load in ad hoc networks. The problem of using PDS model lies in the additional random waiting time for every frame due to medium access mechanism (CSMA/CA in IEEE 802.11). Thus render network scheduler inefficient in providing proportionality between classes. Furthermore, the medium access mechanism was initialy proposed to provide bandwidth fairness between contended nodes, but it is unfair to penalize nodes forwarding more traffic for others with a higher delay than its vicinity. To overcome these problems, and to provide consistent delay at all nodes in the path regardless of their arrival rates and their backlogged traffic, our proposed solution uses a dynamic priority adaptor, Waiting Time Priority (WTP) scheduler at the network layer and a contention window adaptor for IEEE 802.11e. The qualitative and quantitative study of our scheme is conducted after a formal description, expressed through stochastic extensions of process algebras and by simulation through a queueing network model. The algebraic description is not described in this paper due to space limitation.

The rest of this paper is organized as follows. Section II gives a brief introduction to the PDS model with the impediments that prevent its use in ad hoc networks. Section III presents the components of our extended PDS (EPDS). Section IV is devoted to the performance evaluation and analysis. Finally, section V concludes the paper with a summary of the results and future directions.

II. The Proportional Differentiation Service model and properties

There are two basic types of service differentiation schemes [11]. The first one is the absolute service differentiation, which has a weak ability of adaptation to fluctuating arrival rates from various hosts and which leads to a low resource utilization. The second one is relative service differentiation, where QoS measures for a class are guaranteed relatively to others in the network.

Within the relative service differentiation infrastructure, traffic is divided into N classes that are sorted in an increasing order according to their desired levels of QoS. In this scheme, service quality of class i is better than class $i-1$ for $1 \leq i \leq N$. Thus assures that the class with higher priority will receive a relatively better quality than the classes with lower ones. Therefore, the application must regulate its priority levels to meet its end-to-end requirement in an adaptive manner.

The primary objective of relative service differentiation is to provide proportional differentiated level of QoS to different traffic classes even in the presence of burst in a short timescales. The best known effort in the PDS model was initially proposed in [18], [20], [21], which attempts to provide proportional queueing delay differentiation for packet forwarding in IP networks. It states that the average delay examined by classes should be proportional to the differentiation parameters:

$$\frac{d_i(t,t+\tau)}{d_j(t,t+\tau)} = \frac{\delta_i}{\delta_j},\ \forall\, i \neq j\ and\ i,j \in \{1,2,\ldots,N\} \qquad (1)$$

The class parameters δ_i, δ_j are the pre-specified differentiation parameters for class i and j respectively. They are ordered as that higher classes provide lower delay, i.e. $\delta_1 > \delta_2 > \ldots > \delta_N > 0$. $d_i(t,t+\tau)$, $d_j(t,t+\tau)$ are the average delay for class i, j in the time interval $[t,\ t+\tau]$.The delay perceived by each class is relative to another class, and the higher class will get a better service (i.e. lower delay) than lower classes. Equation 1 must hold for each class regardless of its loads, which mean

that the ratio between classes will remain constant depending only on the pre-defined differentiation parameters.

As far as the design for scheduling algorithms to provide proportional delay differentiation, many schedulers have appeared to achieve this proportionality, e.g. waiting time priority (WTP[18]), Proportional Average Delay (PAD [21]) and the Hybrid Proportional Delay (HPD [18]). The difference between these schedulers is related to their speed of convergence under heavy and light load in the network, but it should be noted that all three schedulers use time dependent queueing delays to assign priorities to packets in different fashions.

Furthermore, Dovrolis et al in [20] introduced a method to provide an absolute guarantee to the end user through the proportional differentiated services by adding a dynamic class selection mechanism, where the application can increase/decrease its traffic class dynamically based on the QoS feedback reports from the receiver to satisfy its requirements.

In the rest of this paper, we will adopt the WTP algorithm which was studied in [22] with the name of Time Dependent Priority (TDP), and which was used by Dovrolis in [21] as an effective means to achieve the proportional delay differentiation in IP network. In WTP, the classifier adds $t_{arrival}$ to the header of each packet and forwards it to the corresponding queue according to its belonging priority class. The scheduler serves packets from queues in the FIFO manner by calculating the waiting time of each of head of line (HOL) packet (denoted by $w_p(t) = t - t_{arrival}$) in each queue and chooses the packet with the higher associated priority given by the following formula:

$$p_i(t) = \frac{w_p(t)}{\delta_i} = \frac{t - t_{arrival}}{\delta_i}$$

The scheduler selects packet with the largest priority value at time t from the HOL packets of all backlogged classes to be forwarded, according to the following formula:

$$ser\_p(t) = \arg_{i=1\ldots N} \max(p_i(t))$$

Where N is the set of all backlogged classes. If two packets have the same priority value at time t then they will be transmitted in a random order, but the arrival process of traffic usually follows the Poisson distribution probability where the Pr(2 packets arrive at the same instant) is zero.

The different classes must have an equal waiting time priority at the same node in order to make the required proportionality between classes hold, e.g. transmitted packets of class i and j at time t_1 and t_2 must have:

$$\frac{w_i(t_1}{\delta_i} = \frac{w_j(t_2}{\delta_j} \quad \forall i, j \in \{1, 2, \ldots, N\}$$

While this mechanism is suitable for wired networks, it is still desirable to use this model in the wireless domain. Due to the fact that WTP is a centralized scheduling scheme, it needs to know the waiting times of all packets before deciding which one to transmit at a time. This is trivial in IP network, where all packets waiting to be scheduled originate from the same routers.

Due to shared medium and distributed access mechanism in ad hoc networks (CSMA/CA used in IEEE 802.11 [23]), WTP can not achieve proportionality between classes at the same node, because of the contention based access and the additional random probabilistic waiting time. In contrast to IP networks where the link is controlled by one router, frames underlying different classes at the MAC layer in ad hoc, wait for an additional random time before transmission (discrete uniform random variable), and thus render PDS inefficient with these kinds of networks. This additional random time may cause priority reversal at transmission time, because at this instance, the frame may no longer be the corresponding one to the paquet with the highest priority. For clarification, if packet $Packet_n$ received at MAC layer at time t_n, it will not be transmitted immediately but after a uniformly distributed random time. At transmission time t_{tx}, this packet may no longer have the largest waiting time priority $p_i(t)$ as shown in figure 1.

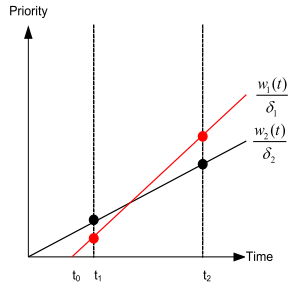


Fig. 1. Priority reversal

Most of the existing distributed QoS schemes for ad hoc provide service differentiation by proposing a new medium access mechanism, e.g. the use of priority based channel access [24], [25], or distributed fair scheduling [26], or linear mapping from network to MAC access priority [27],etc. However, even if some works have shown that it can achieve a relative service differentiation, where higher priority classes have better performance than lower priority classes, they did not provide a means to adjust the degree of differentiation between service classes, nor a formal proof for the differentiation result.

Recently, the IEEE Task Group proposes the Enhanced Distributed Coordination Function (EDCF) in IEEE 802.11e [28], which enhances IEEE 802.11 DCF with the introduction of different traffic classes by the use of distinct Arbitration Inter Frame Spaces ($AIFS_i$) and contention window (CW_i) sizes for different classes. We will exploit this access mechanism with the Markov analysis given in [29] to provide differentiation between classes.

III. Specification of the Proposed Model

Our objective is to extend PDS model in order to provide end-to-end proportional delay differentiation between classes in ad hoc networks. Our proposed model is constructed by

composition of many mechanisms: priority adaptor, proportional differentiation scheduling mechanism, enhanced distributed prioritized medium access EDCF of IEEE 802.11e and delay estimator component as shown in figure 2.

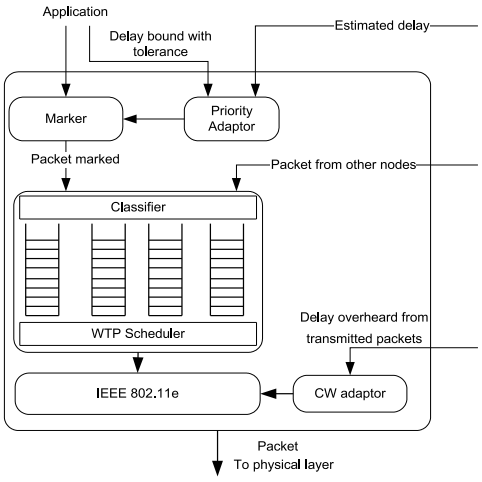


Fig. 2. QoS assurance mechanisms at each node

This mechanism works as follows: each application specify its QoS requirements parameters (maximum supported end-to-end delay, tolerated jitter, minimum throughput, etc.) to the priority adaptor component. This last is used to specify the appropriate priority dynamically in order to meet the QoS application requirements for its flow. The WTP scheduling mechanism is used at the network layer to provide differentiation between tagged packets in the same manner as in IP networks and to forward packets to the IEEE 802.11e layer. At link layer, a modification to the initialization fashion of control parameters in the MAC layer are proposed to provide proportionality between different access categories and packet delay fairness in each class at every node. This modification will be achieved through the use of congestion window adaptor component. In the next sub-section, we give a detailed specification of the tasks of each of these components.

A. IEEE 802.11e and contention window adaptor

In ad hoc networks, nodes access the medium with a decentralized scheduling scheme such as the distributed coordination function (DCF [23]) of IEEE 802.11. It is based on Carrier Sense Multiple Access Collision Avoidance mechanism (CSMA/CA) with binary exponential backoff algorithm when collision occurs.

EDCF in IEEE 802.11e is an extension to DCF mechanism for supporting QoS differentiation. The EDCF basic access method [28] is shortly summarized as follows: each packet from the higher layer arrives at the MAC layer with a specific priority value. A 802.11e station implement four access categories (ACs), where each packet arriving at the MAC layer with a pre-defined priority (traffic categories) is mapped into the corresponding AC. Basically, EDCF uses different Arbitration Interframe Spacing ($AIFS(AC_i)$), minimum Contention Window value ($CW_{\min}[AC_i]$) and maximum Contention Window value ($CW_{\max}[AC_i]$) for differentiation between packets belonging to the different ACs in contention phase to access the channel, instead of single $DIFS$, $CW_{\min}$, and $CW_{\max}$ values as in 802.11 DCF. These parameters will be exploited to provide proportional differentiation between ACs and consistent equal delay for each classes at every node.

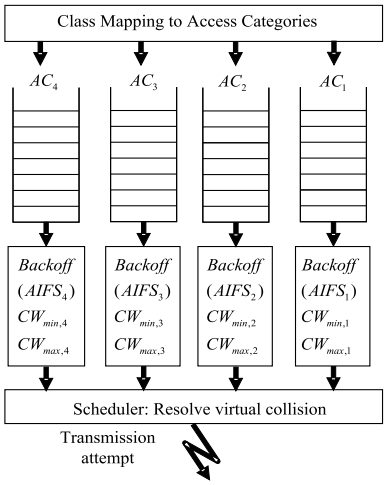


Fig. 3. Four ACs for EDCF.

Figure 3 shows the 802.11e MAC with four transmission queues in a station, where each queue behaves as a single enhanced DCF contending entity, i.e., an AC_i with its own $AIFS(AC_i)$ and Backoff Timer ($BT(AC_i)$). After sensing the channel idle for time period equal to $AIFS(AC_i)$ by an AC_i, it generates a random backoff value before transmitting. The backoff time counter is decremented $BT(AC_i) = BT_{old}(AC_i) - 1$ as long as the channel is sensed idle for unit time ($aSlotTime$). The value of $BT(AC_i)$ is freezed when a transmission is detected on the channel, then reactivated when the channel is sensed idle again for more than $AIFS(AC_i)$. The AC_i transmits when the backoff time $BT(AC_i)$ reaches zero. Moreover, the backoff timer is generated as $BT_i = random(0, CW_{i,j}) \times aSlotTime$, where $random()$ is a generator of random uniformly distributed from $[0, CW_{i,j} - 1]$ interval, and $aSlotTime$ is a very small time period (9μs) and $CW_{i,0} = CW_{\min}(AC_i)$ at the first attempt. After each unsuccessful transmission, $CW_{i,j}$ is increased exponentially by a factor 2 up to a maximum value $CW_{\max}(AC_i)$ as shows equation 2.

$$CW_{i,j} = \left\{ \begin{array}{ll} 2^j CW_{i,0} & 0 \leq j \leq m \\ 2^m CW_{i,0} & m \leq j \leq R \end{array} \right\} \quad (2)$$

R is the retransmission limit at the MAC layer and it is equal to 7 in both DCF and EDCF. $CW_{i,j}$ denotes contention window of class i after a number of unsuccessful transmission j. After, a successful transmission, $CW_{i,j}$ will be reset to $CW_{i,0}$. When more than one AC_i within a station have their $BT(AC_i)$ expire at the same time, the collision is handled in a virtual manner. The highest priority packet among the colliding packets is chosen and transmitted, and the other queue performs the backoff mechanism while increasing $CW(AC_i)$ values.

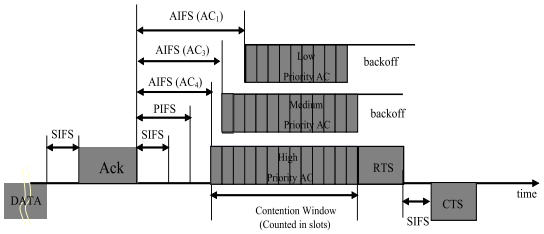


Fig. 4. EDCF channel access mechanism.

The basic access medium in EDCF is shown in Figure 4. This figure shows the timing diagram of the EDCF channel access. Basically, the smaller $AIFS(AC_i)$, $CW_{\min}[AC_i]$, and $CW_{\max}[AC_i]$, the shorter the channel average access delay for the corresponding priority, and hence the more capacity this priority obtains. However, the probability of collisions increases when operating with smaller $CW_{\min}[AC_i]$.

As Malli et al. in [30] show through simulations, EDCF performs poorly when the medium is highly loaded. This is due to the high collision rate and wasted idle slots caused by backoff in each contention cycle. The best medium distribution can be obtained only when EDCF supports a perfect scheduling algorithm among all the queues even in those at different nodes. That involves a complete synchronization and is difficult to realize in this kind of networks.

IEEE 802.11e achieve distributed priority scheduling by moderating the contention behavior in enhanced distributed coordination function (EDCF) [28] where better delays are provided using higher medium access priorities. The service differentiation is qualitative and does provide neither any specific delay assurance nor proportionality between classes. Consequently, we study the influence of the control parameters ($CW_{\min}[AC_i]$, $CW_{\max}[AC_i]$, $AIFS(AC_i)$, etc.) that affect PDS. As a result, we knew that $CW_{\min}[AC_i]$ is the most effective parameters in providing a relative differentiation between classes. Therefore, we should search the relationship between delay constraint and $CW_{\min}[AC_i]$ for providing a consistent differentiation.

Bianchi in [29] presents a Markov chain model for the analysis of the IEEE 802.11 saturation throughput. By analyzing the chain in the way proposed by Chatzimisios et al in [31], we get all required value for average transmitting delay experienced by contending nodes:

$$D_i = \sum_{j=1}^{R} P_{i,j} T_{i,j}$$

where $P_{i,j}$ is the probability that a node i transmits the frame at the $j-th$ backoff stage, and $T_{i,j}$ is the average delay. These parameters are given by Liqiang et al in [32], with:

$$P_{i,j} = (1 - P_i)(P_i)^j \qquad 0 \le j \le R.$$

and

$$T_{i,j} = AIFS_i + \frac{S_i}{2}.\sum_{l=0}^{j}(W_{i,l} - 1) + j.P_i.T_{i,c} \qquad 0 \le j \le R.$$

with S_i is the normalized throughput, P_i is the collision probability and $T_{i,c}$ is the collision time. Interested reader must refer to [31] for a detailed explication. Consequently, the ratio of the average delay of adjacent classes can be written as:

$$\frac{D_{i+1}}{D_i} = \frac{\sum_{j=0}^{R=7} P_{i+1,j} T_{i+1,j}}{\sum_{j=0}^{R=7} P_{i,j} T_{i,j}} =$$

$$\frac{\sum_{j=0}^{R=7}\Big(AIFS_{i+1} + \frac{S_{i+1}}{2}\sum_{b=0}^{j}(CW_{i+1,b} - 1) + \dots}{\sum_{j=0}^{R=7}\Big(AIFS_i + \frac{S_i}{2}\sum_{b=0}^{j}(CW_{i,b} - 1) + \dots} \frac{\dots + j \cdot p_{i+1} \cdot T_{i+1,c}\Big) \cdot P_{i+1,j}}{\dots + j \cdot p_i \cdot T_{i,c}\Big) \cdot P_{i,j}}$$

As shows the previous formula, the control parameters that affect the delay experienced by a packet are: $AIFS_i$, R, $j.p_i.T_{i,c}$, $P_{i,j}$ and the minimum contention window size through $CW_{i,\min}$. Yang and Kravets in [33], minimum contention window sizes and throughput have been shown to have the following relationship, where we can conclude the equivalence through the relation between throughput (TH_i) and transmission delays (D_i):

$$\frac{TH_{i+1}}{TH_i} \approx \frac{\frac{L_{i+1}}{CW_{i+1,\min}}}{\frac{L_i}{CW_{i,\min}}} \Leftrightarrow \frac{D_{i+1}}{D_i} \approx \frac{CW_{i+1,\min}}{CW_{i,\min}} \qquad (3)$$

This can be explained by the fact that p_i is the same for all the classes, $AIFS[i]$ and $j.p_i.T_{i,c}$ are smaller than the other term. Therefore, delay proportionality can hold between classes at the same nodes as in end-to-end, because the packet's end-to-end delay equals the sum of all per-hop delays along the path. On the other hand, it is unfair to penalize nodes relaying more traffic than others. We turn our attention to provide a fair delay for each class at different nodes along the path. We want to provide an extension to equation 1 that must hold locally, to make it hold along every node in the networks, according to:

$$\frac{d_i^p(t, t+\tau)}{d_j^q(t, t+\tau)} = \frac{\delta_i}{\delta_j}, \quad \forall\, i \neq j \; and \; \forall\, p \neq q \qquad (4)$$

The superscripts p and q represent the *id* of two nodes along the path. An equal time delay between contending nodes can be achieved through a dynamic adjustment of minimum contention window as follows:

$$CW_i^p(t_k) = CW_i^p(t_{k-1}) \times (1 + \eta \frac{\bar{d}_N^p(t_{k-1}) - \bar{d}_i^p(t_{k-1})}{\bar{d}_N^p(t_{k-1})}) \quad (5)$$

with $\overline{w_i^p}(t) = \frac{t_{tx}-t_{arrival}}{\delta_i} = \frac{W^p_{packet}}{\delta_i}$ is the normalized delay experienced by a packet at node p, and $\overline{d_i^p}(t_k)$ is the average of this normalized delay calculated as follows:

$$\overline{d_i^p}(t_k) = \alpha \overline{w_i^p}(t_k) + (1-\alpha)\overline{d_i^p}(t_{k-1})$$

$$\overline{d_N^p}(t_k) = \delta \overline{d_i^p}(t_k) + \beta \overline{d_N^q}(t_k) + (1-\delta-\beta)\overline{d_N^p}(t_k)$$

$\bar{d}_N^p(t_k)$ denotes the estimated normalized delay of the network at node p and η is a small positive constant. Each node must estimate its average waiting time $\overline{d_i^p}(t)$ after the transmission of each packet using the RTT average formula, and the average waiting time of the network $\overline{d_N^p}(t)$ after overhearing of a packet transmitted in its contending zone. Clearly, RTS/CTS/DATA/ACK frames can piggyback the waiting time of each packet and the average network estimation delay for two reception zone away along the path. This information in the packet header may be used to estimate the average delay experienced by a packet in the previous hop of the network, and at the node forwarding other flows in the same reception zone. The basic idea is to equalize the average normalized delay between nodes along the path such that equation 4 is satisfied. Therefore, the contention window adaptor must adjust the minimum contention window accordingly, by comparing the average delay of its transmitting packets with the networks average delay estimated from collected data. To provide proportionality at the same node, contention window adaptor updates the minimum contention window of only one predefined class according to equation 5 after a successful transmission, and for other classes according to equation 3.

B. Network layer and waiting time priority scheduler

The classifier handles the received packets by forwarding them to the appropriate waiting queue, where they wait before transmission to the MAC layer. The WTP scheduling is used to provide differentiation at the network layer in the same manner as in IP networks, where packets are treated in a proportional manner. WTP selects the packet with the longest normalized waiting time and sends it to the MAC layer.

C. The Priority adaptor

The end-to-end delay (throughput) sensitive applications request a bounded maximum delay (minimum throughput) with a jitter bound (tolerance bound). At the source node, this mechanism is responsible for determining the suitable class for each traffic flow. It begins by tagging packets with the lowest priority and compares received QoS report feedbacks with the required one. If QoS parameters are not satisfied, it increments the priority by one until the perceived QoS is satisfied or stays in the same class if it reaches the maximum priority level N. The priority adaptor may select directly the adequate priority if there is available information from existing flows. We do not claim to provide a hard guarantee with this priority adaptor, which tries to meet required bounds without providing any guarantee if the network is not able to deliver requirements of the application. This mechanism begin with priority $C_i^0 = 0$ and increases the priority of the flow periodically after T time if the received feedback report is inadequate with applications requirements. This mechanism works as follow:

$$\begin{cases} C_i^0 = 1 & \\ C_i^{k(T+1)} = C_i^{kT} + 1 & if\ (C_i^{kT} < N \wedge QoS_{par} \notin SAT) \\ C_i^{k(T+1)} = C_i^{kT} & if\ (C_i^{kT} = N \wedge QoS_{par} \notin SAT) \\ C_i^{k(T+1)} = C_i^{kT} - 1 & if\ (C_i^{kT} > 1 \wedge QoS_{par} \in SAT) \\ C_i^{k(T+1)} = C_i^{kT} & if\ (C_i^{kT} = 1 \wedge QoS_{par} \in SAT) \end{cases}$$

Where i is a flow indicator and SAT is the satification set of QoS parameters.

IV. Performance Evaluation

In this section we study the performance of the proposed scheme using simulations performed in the $QNAP$. Queueing model is used due to its flexibility in adding time to header of each packets and its offred facility in accessing HOL packets information from other queues. The formal specification of each component in our scheme has been described through the use of algebraic operators of architectural description langage $AEMILA$ [34], before the description in $QNAP$.

We have chosen a a small grid topology of (3×3) (figure 5), with linear mobility in the four directions for all nodes, except A and C supposed fixe. The destination and the sources from where the data have to be sent are randomly generated in addition to that from A to C. When node B fails for an exponential delay used to simulate mobility and link break when node moves out, existing traffic from A to C will travel through AE along the path AEC to reach required destination.

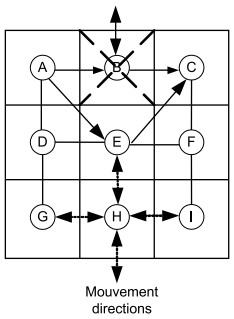


Fig. 5. Topology used in simulation.

We briefly describe the experimental setup and system configuration of our proposed model. Next, we present the results demonstrating the effectiveness of the proposed model in providing proportionality.

The robustness of our proposed EPDS scheme is tested using two different packet arrival profiles. Packets can belong to four different classes and usually the arrival process is described as a Poisson process. This process is the most widely popular traffic model because it takes into account the fluctuation of traffic. The time t between arrivals ($inter - arrival$) is exponentially distributed with rate λ:

$$Pr(t \leq T) = 1 - e^{-\lambda t}$$

and the number of arrivals in an interval of length t is then given by the Poisson probability:

$$Pr(n\ arrivals \in [0,t]) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

In contrast, recent studies and measurements show that realistic traffic follows heavy tailed distribution where the variance of data size is very large, even sometimes not finite and that can not be represented by Poisson distribution. Heavy tailed distributions are more convenient, e.g. Pareto distribution function given in equation 6 is an example of heavy tailed distribution. However, a robust model should not depend at distribution load assumptions for providing QoS.

$$Pr(t \leq T) = 1 - \frac{1}{t^{\kappa}} \quad (6)$$

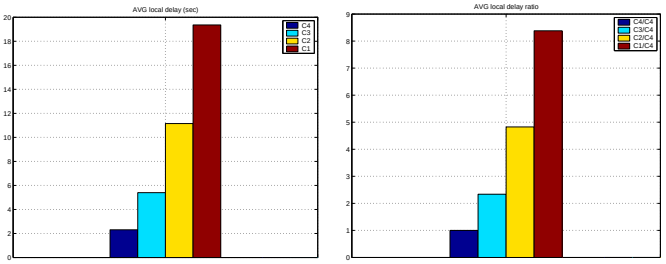
Therefore, we consider Pareto traffic arrivals for each class, where the packet arrival process follows the Pareto distribution with a shape parameter equals to $\kappa = 1.2$. All packets are constant length with 512bytes.

We first study the accuracy of EPDS model in providing differentiation between classes according to the pre-specified ratios at the same node and under the two arrivals pattern. We focus on scenarios of only four service classes at the network layer mapped directly to the 4 access categories used in IEEE 802.11e at MAC layer. All the parameters investigated in our model are given in table I. Results concerning the local average delay at a node are presented in figure 6, 7. It is obvious from these figures that average delay differentiation is mostly achieved simultaneously between different service classes according to their differentiation weight.

Parameters	*Value*
Number of classes at network layer	4
Number of classes at MAC layer	4
Differentiation parameters δ_i, $i \in \{1,2,3,4\}$	$\delta_1 = 1$, $\delta_2 = \frac{1}{2}$, $\delta_3 = \frac{1}{4}$, $\delta_4 = \frac{1}{8}$
MAC $CW_{i,\min}$, $i \in \{1,2,3,4\}$ for EDCF	[64, 32,16, 8]
MAC $CW_{i,\max}$, $i \in \{1,2,3,4\}$ for EDCF	1024
MacOverhead	28 Bytes
$aSlotTime$	$9\mu s$
$SIFS$	$16\mu s$
$DIFS = SIFS + 2 \times aSlotTime$	$34\mu s$
$AIFS_4$	$DIFS$
$AIFS_i = AIFS_{i+1} + aSlotTime$. $AIFS[4]$, $AIFS[3]$, $AIFS[2]$, $AIFS[1]$	$34\mu s$, $43\mu s$, $52\mu s$, $61\mu s$
Average weights α, δ, β	$\alpha = 0.9$, $\delta = 0.1$, $\beta = 0.1$
Per-class queue size (packets)	512bytes
Propagation delay	$1\mu s$
Delay jitter tolerance ε	20% of application delay

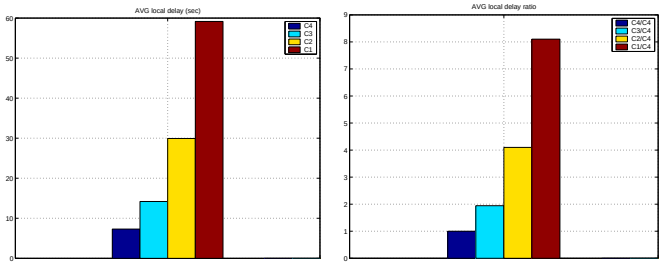
TABLE I
SIMULATION PARAMETERS.

User mobility leads to network topology changes after link breaking and thereby rerouting of all forwarded flows along the old path. When this occurs, traffic distribution changes significantly at other nodes in the same reception zone, and a transient perturbation of the proportionality ratio will occur and thus will result in short timescale violation of proportionality. This perturbation will not appear in the average and therefore a transient study is necessary to detect the influence of mobility at performance degradation. Figures 8, 9 show a non significant perturbation at a local node where proportionality between classes nearly continues to hold in the first 300sec of simulation run. The end-to-end delays proportionality continu hold perfectly with respect to differentiated parameters of $1 : \frac{1}{2} : \frac{1}{4} : \frac{1}{8}$, where we observe that the end-to-end achieved waiting time ratios are significantly closer to the target ratios. Velocity of each mobile node was taken 1m/sec during simulation.



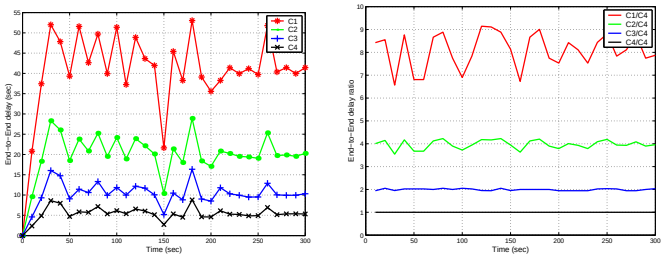
(a) Average local delay. (b) Average local delay ratio.

Fig. 6. Inter-arrival is exponentially distributed.



(a) Average local delay. (b) Average local delay ratio.

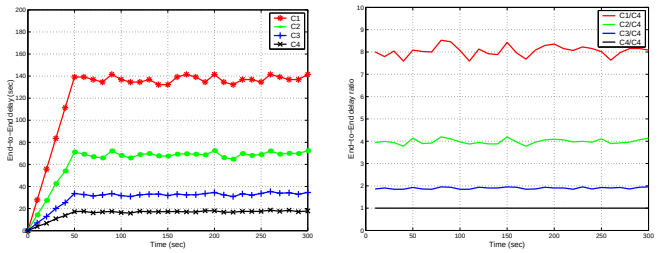
Fig. 7. Inter-arrival is Pareto distributed.



(a) End-to-End delay. (b) End-to-End delay ratio.

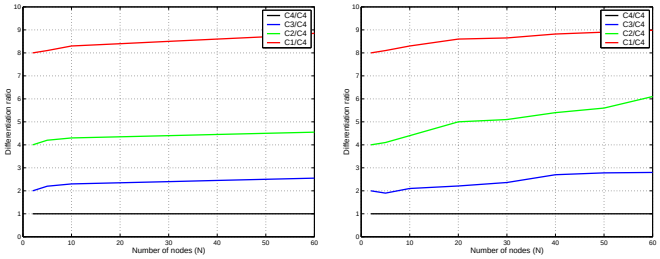
Fig. 8. End-to-End delay and delay ratio with Poisson distribution.

Then we extend the study to the impact of network size N at differentiation between classes. The variation curve is presented in figures 10(a) and 10(b) for Exponential and Pareto



(a) End-to-End delay. (b) End-to-End delay ratio.

Fig. 9. End-to-End delay and delay ratio with Pareto distribution.



(a) Exponential inter-arrival. (b) Pareto inter-arrival.

Fig. 10. Impact of network size.

inter-arrival pattern respectively. We observe that the achieved differentiation ratios are nearly equivalent to their assigned differentiation weights when the network size is small $(N \leq 10)$. In contrast, when the network size is large $(e.g. N \geq 40)$, our scheme tries to maintains a differentiation index close to the target, but it suffers from the number of collision that grows exponentially with the number of nodes (N).

In addition, our model results in a significant performance gain over EDCF that initializes its parameters in a static manner , regardless of channel condition. The gain appears in terms of enhanced throughput, reduced access delai and reduced collision probability even under a large size networks.

V. CONCLUSION

In this paper, we have investigated the problem of delivering high priority packets without over-compromising low priority classes by controlling the quality spacing between different classes. We study the problem of providing proportional delay differentiation by the use of EDCF and PDS. We showed the impact of tuning selected parameters of EDCF mechanism of IEEE 802.11e to provide and maintain service differentiation in the channel.

We investigate the impacts of different arrival pattern rules and show that our proposed scheme has the ability to softly re-adjust bandwidth among different types of traffic, in contrast to current QoS differentiation mechanisms that depend on specific assumptions of the distribution of traffic inter-arrival pattern. Our scheme makes the performance of network adaptively configurable by themselves which will minimize the impact of mobility at performance parameters.

From the performance point of view, we can also observe that our model scheme is an efficient way in providing differentiation between classes in predictable and controllable way. Moreover, our scheme is easy to implement and work in a completely distributed fashion. Finally, it is also possible to incorporate any proportional scheduling mechanism other than Waiting Time Proirity (WTP), to provide better support for differentiated services in mobile ad hoc networks.

REFERENCES

[1] S. Corson and J. Macker, "Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations," RFC 2501, January 1999.

[2] J. Broch, D. B. Johnson, and D. A. Maltz, "The dynamic source routing protocol for mobile ad hoc networks," IETF draft, 16 April 2003.

[3] C. E. Perkins, E. M. Belding-Royer, and S. Das, "Ad hoc on demand distance vector (aodv) routing," RFC 3561, July 2003.

[4] C. E. Perkins and E. M. Belding-Royer, "Quality of service for ad hoc on-demand distance vector," Internet Draft, 14 October 2003.

[5] J. L. R. Leung, E. Poon, A. lot Charles Chan, and B. Li, "Mp-dsr: A qos aware multi path dynamic source routing protocol for wireless ad hoc networks," in *the Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN'01)*, Tampa, Florida, 14–16 November 2001.

[6] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *the Proceedings of MobiCom*. ACM Press, 1998, pp. 85–97.

[7] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *the Proceeding of IEEE Conference on computer communications (INFOCOM97)*, Kobe, Japan, April 1997, pp. 1405–1413.

[8] J. Xue, P. Stuedi, and G. Alonso, "Asap: An adaptive qos protocol for mobile ad hoc networks," in *the Proceeding of 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE PIMRC 2003)*, Beijing, china, 7-10 September 2003.

[9] R. Sivakumar, P. S. Inha, and V. Bharghavan, "Cedar: A core-extraction distributed ad hoc routing algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1454–1465, August 1999.

[10] R. raden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: An overview," RFC 1633,USC/Information Sciences Institute, MIT, Xerox PARC, June 1994.

[11] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, December 1998.

[12] C. Bonnet and N. Nikaein, "A glance at quality of service models for mobile ad hoc networks," in *The proceedings of 16th Congrs DNAC (De Nouvelles Architectures pour les Communications)*, Paris, France, 2-4 December 2002.

[13] I. Chlamtac, M. Conti, and J. N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Elsevier Ad Hoc Networks Journal*, vol. 1, no. 1, pp. 13–64, July 2003.

[14] S. Lee and A. Campbell, "Insignia: In-band signaling support for qos in mobile ad hoc networks," in *the Proceedings of Mobile Multimedia Communications (MoMuC)*, October 1998.

[15] G. S. A. S-B. Lee and A. T. Campbell, "Improving udp and tcp performance in mobile ad hoc networks with insigna," *IEEE Communications Magazine*, pp. 156–165, June 2001.

[16] A. L. Hannan Xiao, Winston K.G. Seah and K. C. Chua, "A flexible quality of service model for mobile ad hoc networks," in *IEEE Vehicular Technology Conference*, Tokyo, Japan, May 2000, pp. 445–449.

[17] A. V. Gahng-Seop Ahn, Andrew T. Campbell and L.-H. Sun, "Swan: Service differentiation in stateless wireless ad hoc networks," in *the Proceedings of IEEE INFOCOM 2002*, June 2002.

[18] P. R. Constantinos Dovrolis, Dimitrios Stiliadis, "Proportional differentiated services: Delay differentiation and packet scheduling," *IEEE/ACM Transactions on Networking (TON)*, vol. 10, no. 1, pp. 12–26, February 2002.

[19] M. K. H. Leung, J. C. S. Lui, and D. K. Y. Yau, "Adaptive proportional delay differentiated services: characterization and performance evaluation," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 801–817, December 2001.

[20] C. Dovrolis, "Dynamic class selection: From relative differentiation to absolute qos," in *ICNP '01: Proceedings of the Ninth International Conference on Network Protocols (ICNP'01)*. Washington, DC, USA: IEEE Computer Society, September 2001, pp. 120–128.

[21] P. R. Constantinos Dovrolis, Dimitrios Stiliadis, "Proportional differentiated services: Delay differentiation and packet scheduling," in *Proceedings of SIGCOMM*, Cambridge, Massachusetts, United States, 1999, p. 109120.

[22] L. Kleinrock, "A delay dependent queue discipline," *Naval Research Logistics Quarterly*, vol. 11, no. 4, pp. 329–341, December 1964.

[23] IEEE, "IEEE Std. 802.11: Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.

[24] X. Yang and N. H. Vaidya, "Priority scheduling in wireless ad hoc networks," in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM Press, 9 –11 June 2002, pp. 71–79.

[25] J. L. Sobrinho and A. S. Krishnakumar, "Real-time traffic over the ieee 802.11 medium access control layer," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 172–187, 1996.

[26] S. Gupta, N. H. Vaidya, and P. Bahl, "Distributed fair scheduling in a wireless lan," in *Sixth Annual International Conference on Mobile Computing and Networking*, Boston, August 2000.

[27] Y. Xue, K. Chen, and K. Nahrstedt, "Achieving proportional delay differentiation in wireless lan via cross-layer scheduling," *Journal of Wireless Communications and Mobile Computing, Special Issue on Emerging WLAN Technologies and Applications*, vol. 4, pp. 849–866, 2004.

[28] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, "Ieee 802.11e wireless lan for quality of service," in *European Wireless*, Florence Italy, 26-28 February 2002, pp. 32–39.

[29] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 3 March 2000.

[30] M. Malli, Q. Ni, T. Turletti, and C. Barakat, "Adaptive fair channel allocation for qos enhancement," in *the proceedings of IEEE International Conference on Communications (IEEE ICC04)*, Paris, France, July 2004.

[31] P. Chatzimisios, V. Vitsas, and A. C. Boucouvalas, "Throughput and delay analysis of ieee 802.11 protocol," in *the Proceedings of the 5th IEEE International Workshop on Networked Appliances*, L. J. M. University, Ed., no. ISBN:0-7803-7686-2, UK, 30–31 October 2002, pp. 168–174.

[32] L. Zhao and C. Fan, "Enhancement of qos differentiation over ieee 802.11," *WLAN IEEE Communications Letters*, vol. 8, no. 8, pp. 494–496, August 2004.

[33] Y. Yang and R. Kravets, "Distributed qos guarantees for realtime traffic in ad hoc networks," in *the Proceedings of the IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON-2004)*, October 2004, pp. 118–127.

[34] A. Aldini and M. Bernardo, "On the usability of process algebra: an architectural view," *Theoretical Computer Science*, vol. 335, no. 2-3, pp. 281–329, 2005.

Intelligent Solution for Congestion Control in Wireless Ad hoc Networks

L. Khoukhi, S. Cherkaoui
Department of Electrical and Computer Engineering, University of Sherbrooke
J1K 2R1, QC, Canada
{Lyes.Khoukhi, Soumaya.Cherkaoui}@USherbrooke.ca

***Abstract*—In this paper, an intelligent solution is proposed for the congestion control in wireless ad hoc network. The presented QoS architecture exploits the fuzzy logic for improving the congestion control in the aim to support voice and video multimedia applications, and non-real-time traffic services. We present two techniques: the first one uses a fuzzy logic system to perform the threshold buffer management. The second technique is based on Fuzzy Petri nets (FPWICC) for modeling and analyzing the QoS decision making for congestion control in wireless ad hoc network. The performance evaluation was studied under different channel, mobility, and traffic conditions. The simulation results confirm that the intelligent fuzzy logic tool is a promising solution to support QoS for multimedia applications in wireless ad hoc networks.**

***Index Terms*—QoS, multimedia application, congestion control, fuzzy logic, threshold management, fuzzy Petri nets.**

I. INTRODUCTION

THE rapid development of the wireless technology has been accompanied by an evolution of new multimedia applications. Sample multimedia applications include videoconferencing, distance learning, distributed games, video on demand, etc. These applications consist of voice and video traffic, and they need delay and loss guarantees. Others applications, such as World Wide Web and FTP, are delay-insensitive. However, maintaining the end-to-end voice and video quality is too difficult to be accommodated by the wireless ad hoc networks.

Wireless ad hoc networks are self-creating, self-organizing, and self-administering [1]. The dynamic nature of these networks presents significant technical challenges to ensure better quality delivery to the multimedia applications. These applications generate traffic at varying rates and usually require that the network be able to carry traffic at the rate at which they generate it. Besides, they are more or less tolerant in terms of traffic delays and variation in traffic delay. Therefore, it is vital to support the Quality of Service (QoS) for multimedia applications.

In response to the challenges posed by the wireless ad hoc technology, various approaches and protocols have been proposed [4]-[15]. Note that the classical QoS architectures (e.g., IntServ [2] and DiffServ [3]) proposed for the wired networks are not readily applicable to this new technology. Recently, many researches have focused on addressing the QoS issue in the ad hoc networks: SWAN [4], INSINGIA [5], and FQMM [6]. The later model is a hybrid approach combining the advantages of per-class granularity of DiffServ with the per-flow granularity of IntServ. It tries to preserve the per-flow granularity for a small portion of traffic in MANETs, given that a large amount of the traffic belongs to per aggregate of flows, that is, per-class granularity. FQMM offers a good solution for small- and medium-size ad hoc network, but it is not suitable for large networks. INSIGNIA is one of the noteworthy QoS frameworks with per-flow granularity and reasonable treatment for mobility. The main goal of INSIGNIA is to provide adaptive QoS guarantees for real-time traffic. It employs an in-band signaling system that supports fast reservation, restoration, and adaptation algorithms. Three levels of services are implemented: best-effort, minimum, and maximum. The bandwidth is the only QoS parameter used in INSIGNIA. SWAN proposes a service differentiation in stateless wireless ad hoc networks by using distributed control algorithms. It relies on feedback from the MAC layer as a measure of congestion in the network by using a mechanism of rate control and source-based admission control. It promotes a rate control system that can be used at each node to treat traffic either as real-time or best-effort traffic. However, one of the drawbacks of SWAN is how to calculate the threshold rate limiting any excessive delay that might be experienced [7]. SWAN uses merely two levels of services: real-time and best-effort traffic, but it remains the best example of stateless distributed QoS framework developed for wireless ad hoc networks.

We have proposed in [9] GQOS model, which is an intelligent QoS model with service differentiation based on neural networks for mobile ad hoc networks. GQOS is composed of a kernel plan which assures basic functions of routing and QoS support control, and an intelligent learning plan which assures the training of GQOS kernel operations by using multilayered feedforward neural network (MFNN). The advantages of using neural networks algorithm is the fast learning of different operations performed by the kernel and the reduction of time processing in the network. However, the results of simulation show that GQOS is not suitable for high dynamic networks. To overcome this drawback, we have explored in [10] the use of a fuzzy logic semi-stateless QoS

approach for service differentiation in wireless ad hoc networks, called FuzzyMARS. This architecture support both real-time UDP traffic and best-effort UDP and TCP traffic. The resulted simulations have shown the benefits of using fuzzy logic semi-stateless model, the average delay obtained is quite stable and low under different channel conditions, traffic scalability, and mobility scenarios. Nevertheless, in FuzzyMARS we did not consider buffer management.

In this paper, we propose a new QoS approach for wireless ad hoc networks, named FuzzyCCG. This approach explores the fuzzy logic for improving the control of congestion for multimedia applications. FuzzyCCG exploration is useful first, because of the dynamic nature of buffer occupancy and congestion at a node, second, because of the uncertain nature of information in wireless ad hoc networks due to the network mobility. FuzzyCCG proposes to use fuzzy logic approach for threshold selection in order to deal with the dynamic buffer occupancy and the uncertain and imprecision nature of wireless ad hoc network information. Using fuzzy logic, FuzzyCCG investigates the fuzzy thresholds ability to adapt to the dynamic conditions over the classical inflexible thresholds. The notion of threshold is practical for discarding data packets and adapting the traffic service depending on the occupancy of buffers. Therefore, the selection of a particular threshold may be decisive to the control of congestion, and therefore to the network performances.

This paper proposes also a new fuzzy Petri nets technique, named FPWICC. The objective is to model and analyze the QoS decision making for congestion control in wireless ad hoc networks. The fuzzy Petri nets tool is used for its efficiency and flexibility over other modeling tools (such as Petri nets) in the aim of better modeling and representation the process of buffer management.

The performances of FuzzyCCG are studied under different network conditions in terms of network conditions. The simulations results shown in Section IV confirm that the proposed approach offers promising results to support multimedia applications.

The rest of the paper is structured as follows: Section II describes the proposed architecture. Section III illustrates the proposed fuzzy Petri model for congestion control. The simulation results under different network conditions are shown in Section IV. Finally, Section V concludes the paper.

II. FUZZYCCG ARCHITECTURE

A. Overview of Fuzzy logic

L. Zadeh has introduced in the 1960s the Fuzzy logic theory [16]-[17] as a tool for modeling the uncertain of natural language, which has been commonly employed for supporting intelligent systems. This technology has proven efficiency in a various applications such as decision support and intelligent control, especially where a system is difficult to be characterized. A fuzzy logic system considers basically three steps: fuzzification, rules evaluation, and deffuzification. The first step is responsible for mapping discrete (called also crisp) input data into proper values in the fuzzy logic space. For that end, membership functions (fuzzy sets) are used to provide smooth transitions from false to true (0 to 1). The second step performs reasoning on the input data by following predefined fuzzy rules. Once the input data are processed by fuzzy reasoning, the deffuzification takes the task of converting back these input data into crisp values.

B. Fuzzy logic approach for threshold management

The choice of applying fuzzy logic is justified by the fact that fuzzy logic is well adapted to systems characterized by imprecision states such as the case of ad hoc networks. Also given the results found with FuzzyMARS [10], fuzzy logic promises to offer an efficient tool for buffer management by using adequate thresholds that deal with the imprecise information in a wireless ad hoc network. Also, fuzzy logic has been successfully applied to the queue management in the cell-switching networks [18]. Nevertheless, to the best of our knowledge, this is the first work that uses fuzzy logic for buffer management in MANETs. We aim to apply a fuzzy technique based on fuzzy sets theory. The later extends the classical logic set {0, 1} to use linguistic variables (e.g. full buffer, merely full buffer, empty buffer).

Using fuzzy logic, we investigate the fuzzy thresholds ability to adapt to the dynamic conditions over the classical inflexible thresholds. The classical thresholds are characterized by their limitation and restriction, because the selection of threshold is based on a single value. Thus, the utilization of a buffer may be either “poor” or “surcharged”. When the selected value is small (e.g. 30% of capacity), then the admission of new packets is possible only when the buffer occupancy is low. This means a poor utilization of the buffer; since most of incoming packets are rejected even if the buffer is almost unfilled. On the other side, when the selected value is big (e.g. 90% of capacity), problems may happen when the bursty traffic is used. The transmission of packets generated by a bursty traffic is very changing. It can vary from small to “near-peack” rate in a short period of time.

Manually predefining a value for threshold in ad hoc network is not suitable because most of events occurring in an ad hoc network are dynamic and random. In addition, it is important to note that the rate of packets arriving on a particular node is not static. The threshold value divides the buffer into an “admitted” part and a “no-admitted” part. Let consider that the threshold of the buffer shown in Fig. 1.a is equal to 60%. In this scheme, the occupancy level may range from 0 to 60%. When the buffer occupancy is superior to 60%, no incoming packets is accepted in the buffer. Therefore, the change in decision making from “admit state” to “no-admit state” is performed from 60-61%. This means that a small variation in the buffer occupancy may influence the decision making of incoming packets.

The aim of introducing fuzzy logic is to develop a more realistic representation of buffer occupancy that helps to offer an efficient decision making. The proposed architecture

attempts to extend the two-discrete states "admit" and "no-admit" of the buffer occupancy by using fuzzy logic. Hence, the definition of "buffer occupancy" will consider the two fuzzy cases of "getting full" and "not getting full", rather than "admit" and "no-admit" in the existing approaches. This fuzzy representation replaces the two-discrete sets by a continuous set membership, and performs small gradual transitions between different states of buffer occupancy.

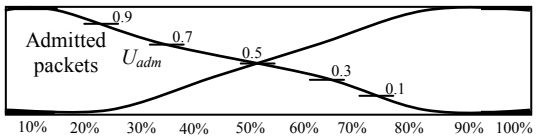
The fuzzy membership function aims to determine the fuzzy threshold based on the fullness of the buffer. For that purpose, several membership functions may be used: "triangular", trapezoidal", or "sigmoid" function. These functions can give a representation about the buffer fullness level. In FuzzyCCG, we used the sigmoid membership function. This choice is based on the fact that this function would reflect well the dynamic occupancy of the buffers that we want to model.

It is observed in Fig. 1.b that the admit membership function is inversely proportional to the occupancy fullness level of buffer. Thus, when the occupancy fullness is small, the value of the admit membership function is big. At higher fullness occupancy levels, the admit membership function value becomes small. When the value of the "no-admit" membership function is getting big, then only a small quantity of packets will be permitted to enter the buffer. In Fig. 1.b, the value of the membership function is represented by the symbol u_{adm}. The fuzzy rules associated are as follows:

<< When the value of the admit membership function is big, then increase the accepted incoming packets into buffer >>.
<< When the value of the admit membership function is small, then reduce the accepted incoming packets into buffer >>.

These fuzzy rules are illustrated by Fig. 1.b. The rejection of packets is controlled based on the degree of fullness of the buffer. For instance, when the buffer is occupied at 40%, this means that the value of u_{adm} is about 0.7 (i.e. the amount of packets admitted is about 70%). Then, about "30%" of incoming packets will be not admitted. Note that the fuzzy threshold approach covers the continuous set of values representing possible buffer occupancy (i.e. from 0 to u_{adm}). This is opposite to the classical threshold approaches that hold only one predefined single value. Therefore, fuzzy logic adds more flexibility to the threshold selection.

1.a) Classical buffer scheme



1.b) Fuzzy Buffer occupancy scheme

Fig. 1. Classical and fuzzy buffer schemes

III. Congestion Control Model Based on Fuzzy Petri Nets

In the previous sections, we presented the fuzzy logic approach for the congestion control. We interest in what follows, to model and analyse the QoS making decision for congestion control using a Fuzzy Petri tool. In [27], we have presented a similar model for the traffic regulation.

The proposed model, called FPWICC, studies the fuzzy congestion control rules in order to deal with the imprecise information caused by the dynamic topology of ad hoc networks. The representation of different fuzzy processes for decision making can be performed by formulating the production rules of these processes. Each fuzzy production rule is a set of antecedent input conditions and consequent output propositions. We proceed to construct the previous aspects (the input and output parameters) of the production rules in order to better represent and understand the process of congestion control in wireless ad hoc networks. The congestion control process used to avoid the congestion depends on the buffer occupancy and the dynamic topology of the network. These constraints represent the input parameters of FPWICC. The buffer occupancy is given the FuzzyCCG admit membership function. The later can give information about the status of a network in terms of congestion. A small value of this parameter signifies that congestion may be appeared in the network. Therefore, the process of traffic regulation should be started. The amount of the accepted incoming packets into buffer represents the output parameter of FPWICC. The choice of using fuzzy Petri nets tool is due to its efficiency and flexibility over other modeling tools for better representing the congestion control process.

A. Fuzzy Petri Nets

It is observed that classical Petri Nets [19] do not have sufficient capacity to model the uncertainty in systems [20]. This limitation of Petri nets has encouraged researchers to extend the exiting models by using the fuzzy reasoning theory [21] [22]. The combination of Petri nets models and fuzzy theory has given rise to a new modeling tool called Fuzzy Petri Nets (FPN). FPN formalism has been widely applied in several applications such as, robotics systems [23], and real-time control system [20], fuzzy reasoning systems [25], etc…

A brief description about the FPN modeling tool is presented in the following [22] [24].

Let consider FPN = (PN, CND, MF, FSR, FM).

a. The tuple PN = (P, T, A, FW, FH) is called Petri nets if: (P, T, A) is a finite net, where [19]:
   $P = \{P_1, P_2, \ldots, P_n\}$ is a finite non-empty set of places,
   $T = \{T_1, T_2, \ldots, T_n\}$ is a finite non-empty set of transitions,
   $A \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs between the places and transitions or vice versa.
   FW: $A \rightarrow N^+$ represents a weighting function that associates with each arc of PN a non-negative integer of N^+.

$FH \subset (P \times T)$: represents an inhibition function that associates a place $P_i \in P$ contained in FH (T_j) to a transition T_j itself.

b. CND = {cd_1, cd_2, …., cd_n} represents a set of conditions that will be mapped into the set P; each $cd_i \in CND$ is considered as one input to the place $P_i \in P$. A condition cd_i takes the form of "X is Z", which means a combination between the fuzzy set Z and the attribute X of the condition. For instance, in the condition "the admit membership value is small", the attribute "X = admit membership value" is associated to the fuzzy set "Z = small", but other fuzzy sets can also be considered (e.g. "Z = medium", "Z = large", etc.).

c. Consider MF: $u_Z(x) \to T$, a membership function which maps the elements of X (as defined in b.) into the values of the range [0,1]. These values represent the membership degree in the fuzzy set Z. The element x belonging to X represents the input parameter of the condition "X is Z", and $u_z(x)$ measures the degree of truth of this condition. Note that the composition of membership function degrees of the required conditions is performed by fuzzy operators such as MIN/MAX.

d. Let consider the following rule R_i:

R_i: if x_1 is z_1 and /or x_2 is z_2 then A is B

The firing strength function of rule R_i (FSR_i) represents the strength of belief in R_i. The conclusion of R_i (modeled by CSR_i) can take one of the following forms:

$$CSR_i = MIN(u_{z1}(x_1), u_{z2}(x_2)) = u_{z1}(x_1) \wedge u_{z2}(x_2)$$

$$CSR_i = MAX(u_{z1}(x_1), u_{z2}(x_2)) = u_{z1}(x_1) \vee u_{z2}(x_2)$$

e. SWR is the selected wining rule R_L among the n-rules R_1, R_2, …., R_n. SWR is the rule which has the highest degree of truth. Let FSR_L be the corresponding firing strength of R_L, then the selected rule SWR is given as follows:

$$SWR = MAX(FSR_1, FSR_2, ..., FSR_n)$$

f. The marking task in FPN illustrates the satisfaction of events occurred during the performance of fuzzy rules. This marking function called "fuzzy marking" (FM) distributes the tokens over the places of the nets.

The sequence $\delta = \langle T_1, T_2, ..., T_n \rangle$ is said to be reachable from a fuzzy marking FM_1, if $T_i \in T$ is a firable from $FM_{i-1} \in FM$ and leads to $FM_{i+1} \in FM$, for all transitions $T_i \in \delta$. The firing of transition $T_i \in T$ (Fig. 2) is performed in two steps: a) T_i removes tokens and then, b) T_i places tokens.

B. *Fuzzy Congestion Control Rules Using*

The following form of modeling is used by most of fuzzy systems [26]:

Rule R: if Ip_1 is A AND Ip_2 is B then Op is C

Where:

- Ip_1 and Ip_2 are the input parameters,
- Op is an output parameter,
- A, B, and C are fuzzy sets,
- AND represent fuzzy operator,
- The fuzzy conditions of rule R are "Ip_1 is A", and "Ip_2 is B".

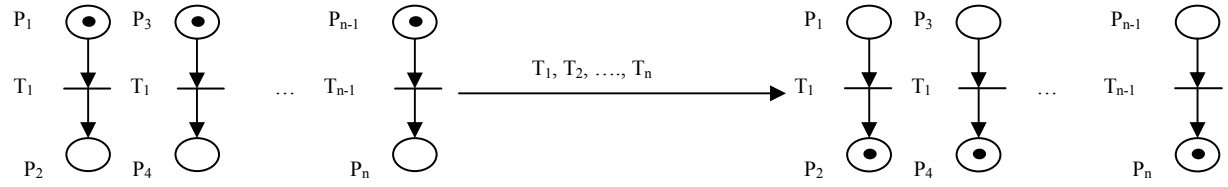


Fig. 2. The transitions firing in FPN

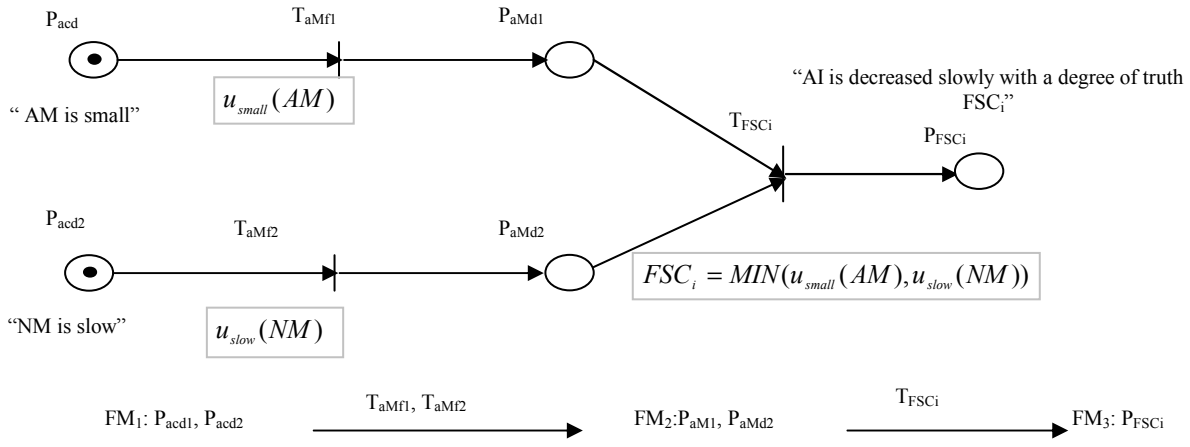
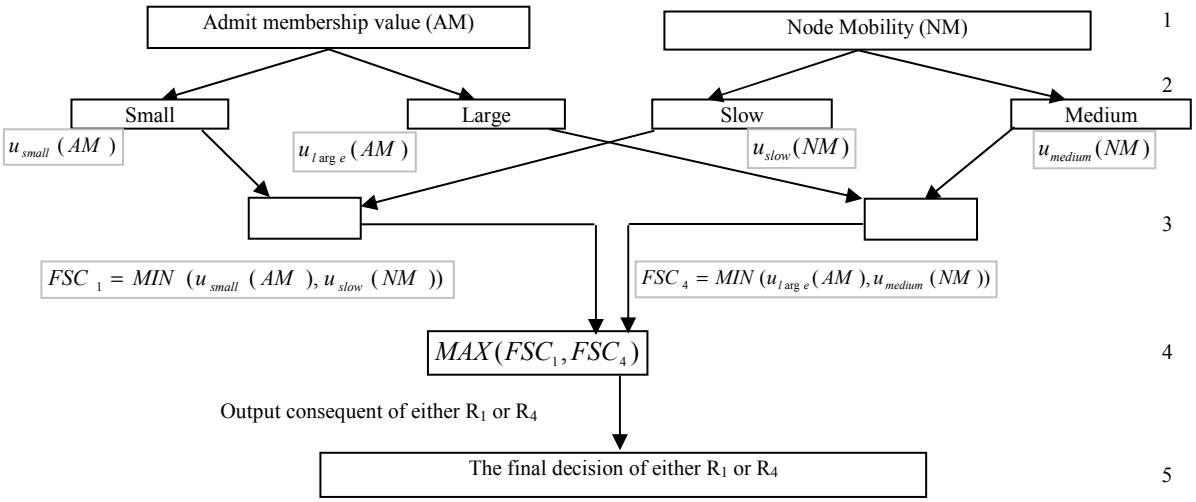


Fig. 3. The modeling of fuzzy rules structure and its dynamic behaviour



Decision making algorithm:

- Phase 1: enter the input parameters of the rules R_1, R_2, R_3, and R_4.
- Phase 2: calculate the degree of truth of the antecedent conditions.
- Phase 3: apply the operation of minimum composition (MIN) with the fuzzy operator AND/OR in order to generate the firing strength value for each rule R_1, R_2, R_3, and R_4.
- Phase 4: apply the operation of maximum composition to select the wining rule among the rules R_1, R_2, R_3, and R_4.
- Phase 5: generate the output consequent of the selected wining rule.

Fig. 4. The fuzzy decision making mechanism of FPWICC

These above aspects (inputs, outputs, and fuzzy sets) are constructed to perform the control of congestion depending on the buffer occupancy and the dynamic topology of wireless ad hoc networks. For that aim, FPWICC uses both the buffer occupancy and the network mobility parameters. Thus, the previous fuzzy aspects can take various values:

- The first input parameter: is represented by the Admit Membership value (AM) at a mobile node. AM can be either small or large.
- The second input parameter: is represented by the Node Mobility (NM). NM can either be slow or medium (note that "fast node mobility" is included in the case of "medium node mobility").
- The output parameter: is represented by the Accepted Incoming packets into buffer (AI). AI can either be decreased (slowly or largely) or increased (slowly or largely).

FPWICC uses the previous rules to help to establish production rules that make an efficient QoS decision. In the following, we explain the proposed fuzzy tool for the QoS decision making.

Let consider the following fuzzy rule R_L:
Rule R_L: if AM is small and NM is slow, then AI is decreased.

R_L takes into consideration the input parameter of the admit membership value AM in the buffer and the node mobility NM in wireless ad hoc networks. The accepted incoming packets into buffer AI represents the output parameter.

FPN that models the dynamic aspect of the fuzzy rule R_L is illustrated in Fig. 3.

- P_{acd1}: models the antecedent condition 1 (acd_1) of R_L; acd_1 = "AM is small".
- P_{acd2}: models the antecedent condition 2 (acd_2) of R_L; acd_2 = "NM is slow".
- T_{aMf1}: models the membership function of the antecedent condition 1; $T_{aMf1=}u_{small}(DM)$.
- T_{aMf2}: models the membership function of the antecedent condition 2; $T_{aMf2=}u_{slow}(NM)$.
- P_{aMd1}: models the membership degree value of the condition 1 of a rule R_L. This value determines the satisfaction degree of the AM input parameter to the fuzzy set "small".
- P_{aMD2}: models the membership degree value of the condition 2 of a rule R_L. This value determines the satisfaction degree of the NM input parameter to the fuzzy set "slow".
- T_{FSCL}: models the operation of minimum composition "MIN" between the antecedent conditions (e.g. condition 1 and condition 2) of a rule R_L. The firing strength of R_L is represented by the MIN operation: $MIN(u_{small}(DM), u_{slow}(NM))$.
- P_{FSCL}: models the value of the firing strength of R_L. This value defines the degree of truth of the output proposition "AI is decreased".

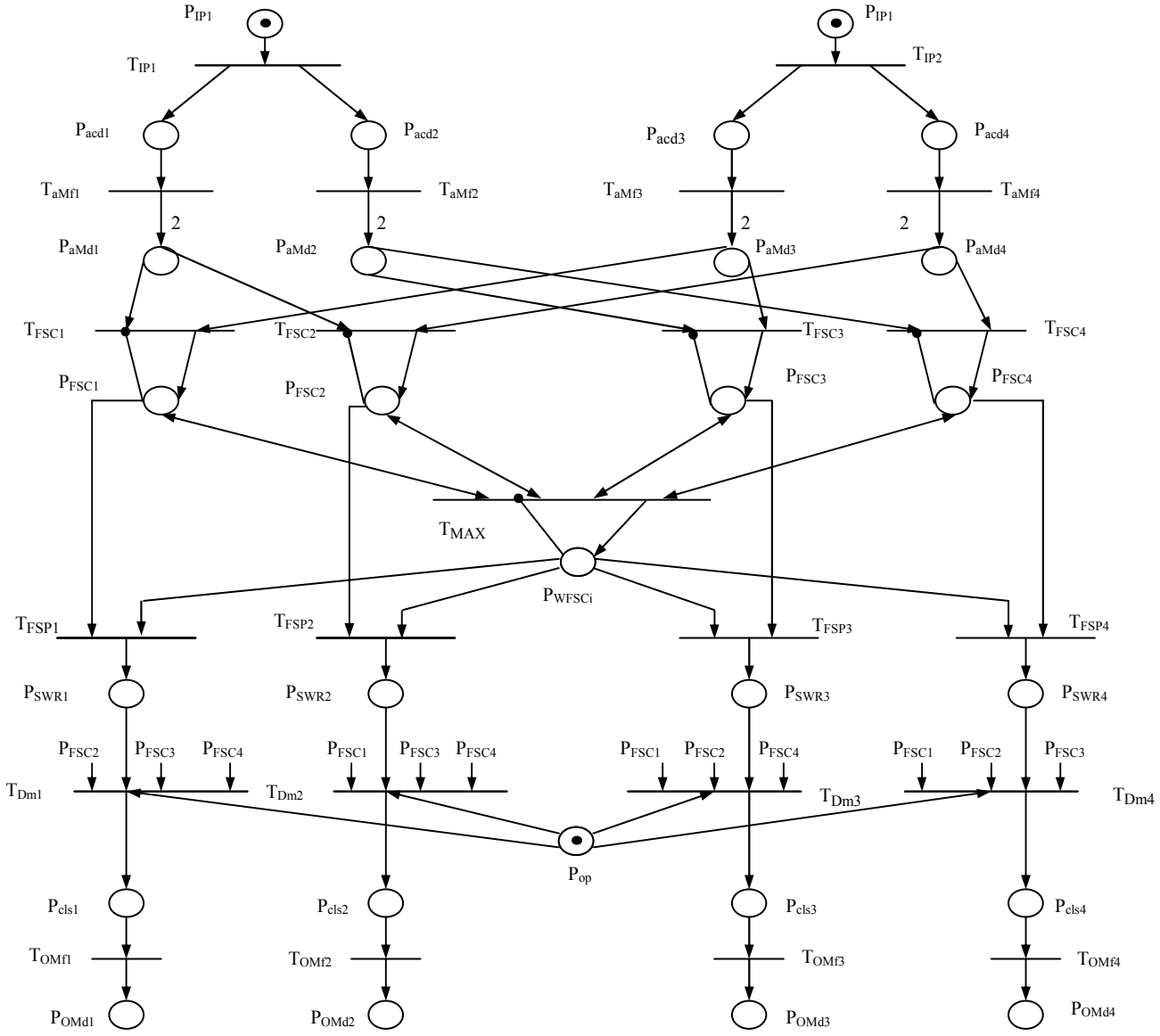


Fig. 5. FPWICC model

C. *Fuzzy Petri Nets Model for Congestion Control*

FPWICC considers the following rules:

R_1: if AM is large and NM is slow then AI is increased largely,

R_2: if AM is large and NM is medium then AI is increased,

R_3: if AM is small and NM is slow then AI is decreased,

R_4: if AM is small and NM is medium then AI is decreased largely.

- Input parameters:
  - The input parameter of the first antecedent condition of the rules R_1, R_2, R_3, and R_4 is the admit membership value AM.
  - The input parameter of the second antecedent condition of the rules R_1, R_2, R_3, and R_4 is the node mobility NM.
- Fuzzy sets:
  The fuzzy set of the antecedent conditions of the defined rules R_1, R_2, R_3, and R_4 are: small, large, slow, and medium.
- Antecedent conditions (acd_i):
  - The first antecedent condition (acd_1) in the rules R_1, R_2, R_3, and R_4 is:
    acd1: AM is small.
    acd2: AM is large.
  - The second antecedent condition (acd_2) in the rules R_1, R_2, R_3, and R_4 is:
    acd1: NM is slow.
    acd2: NM is medium.
- Output parameters:
  The output parameter of the rules R_1, R_2, R_3, and R_4 is the accepted incoming packets into buffer (AI).
- The rules R_1, R_2, R_3, and R_4 use the following decisions making: increased largely, increased, decreased, decreased largely,
- The fuzzy logic operator used by the rules R_1, R_2, R_3,

and R_4 is AND

The fuzzy operator "AND" is used in order to combine the two antecedent conditions of each rule using the MIN function. This provides the firing strength value for each rule. After that, MAX composition function is used to combine all firing strength values of the defined rules R_1, R_2, R_3, and R_4 in the aim of determining the highest one that will be the selected wining rule. Fig. 4 shows the fuzzy logic scheme for decision making of rules R_1, R2, R3, and R4.

In the following, we illustrate the steps of the proposed FPN model.

a. Enter the input parameters into the places and transitions:

- $P_{IP} = \{P_{IP1}, P_{IP2},..., P_{IPn}\}$ is a set of places that represent the input parameters. In the Fig. 5, the places used are P_1 and P_2 which represent respectively, the first (e.g. admit membership value AM) and second (e.g. node mobility NM) antecedent condition of the rules R_1, R_2, R_3, and R_4.
- $T_{IP} = \{T_{IP1}, T_{IP2}, ..., T_{IPn}\}$ represents a set of input parameter transitions. The transitions T_{IP1} and T_{IP2} illustrated in Fig. 5 are used to distribute respectively, the input parameters "AM" and "NM" for making the first and second antecedent conditions of the defined rules R_1, R_2, R_3, and R_4.

b. Represent the antecedent conditions, and compute the membership function for each condition.

- $P_{acd} = \{P_{acd1}, P_{acd2}, ..., P_{acdn}\}$ is a set of places that represent the antecedent conditions. P_{acd1} and P_{acd2} in the model presented in Fig. 5 describe respectively, the antecedent conditions "acd_1" and "acd_2".
- $T_{aMf} = \{T_{aMf1}, T_{aMf2},..., T_{aMfn}\}$ is a set of transitions that represent the antecedent membership functions. T_{aMf1}, T_{aMf2}, T_{aMf3}, T_{aMf4} observed in Fig. 5 represent the membership functions of respectively, $u_{small}(DM)$, $u_{large}(DM)$, $u_{slow}(NM)$, $u_{medium}(NM)$.
- $P_{aMd} = \{P_{aMd1}, P_{aMd2},..., P_{aMdn}\}$ is a set of places that represent the antecedent membership degrees. The values of the place P_{aMd1} indicates the degree of satisfaction of the input parameter AM to the fuzzy set "small".

c. Compute the firing strength of conditions

- $T_{FSC} = \{T_{FSC1}, T_{FSC2}, ..., T_{FSCn}\}$ represent a set of transitions that model firing strength conditions. For instance, the transition T_{FSC1} shown in Fig. 5 performs the operation of minimum composition (MIN) on the antecedent conditions of the rule R_1: $MIN(u_{small}(MD), u_{slow}(NM))$. Note that the fuzzy operator AND is integrated with the MIN operation to combine the first and second conditions of R_1.
- $P_{FSC} = \{P_{FSC1}, P_{FSC2}, ..., P_{FSCn}\}$ is a set of places that represent the firing strength. P_{FSCi} tokens are proportional to the number of antecedent conditions of a rule R_i. This number is shown by the label illustrated between the transitions T_{aMfi} and the place P_{aMdi}. The construction of the antecedent conditions of a rule R_i is performed by firing a transition T_{FSCi}. The inhibitor arc designed between a place P_{FSCi} and T_{FSCi} is useful to note that T_{FSCi} should fire one time.

d. Determine the selected wining rule among the activated rules:

- T_{FMAX} = MAX $\{P_{FSC1}, P_{FSC2}, ..., P_{FSCn}\}$ is a transition that models the maximum composition operation (MAX) for the defined rules. The firing strength value of a rule R_i is stored in the place P_{FSCi}.
- P_{WFSCi} represents the firing strength condition FSC_i of the selected wining rule R_i. The later rule is determined as in the following step.
- $T_{FSP} = \{T_{FSP1}, T_{FSP2},..., T_{FSPn}\}$ is a set of transitions that model the firing strength comparison. For instance, the transition T_{FSP3} is useful to make a comparison between FSC_3 of the rule R_3 and the selected wining firing strength $WFSC_i$.
- $P_{SWR} = \{P_{SWR1}, P_{SWR2}, ..., P_{SWRn}\}$ is a set of places that models the selected wining rules. The rule R_i is selected to be fired if the place P_{SWRi} contains a token.

e. The conclusion of the selected rules:

- $T_{Dm} = \{T_{Dm1}, T_{Dm2}, ..., T_{Dmn}\}$ is a set of transitions that represent the decision of the selected rule. T_{Dmi} deletes the firing strength values of other rules in order to fire only the selected rule R_i.
- P_{op} is a place that models the output parameter. As shown in Fig. 5, the place P_{op} represents the accepted incoming packets into buffer.
- $P_{cls} = \{P_{cls1}, P_{cls2}, ..., P_{clsn}\}$ models a set of places that describe the different decisions of the defined rules. The places P_{cls1}, P_{cls2}, P_{cls3}, and P_{cls4} illustrate the following conclusions respectively, "increased largely", "increased", "decreased", and "decreased largely". Only one place among all places will contain a token which represent the conclusion of the selected wining rule. For instance, the conclusion of the selected rule R_1 is "increased largely" if T_{Dm1} transfers a token from the place P_{SWR1} to the place P_{cls1}.
- $T_{OMf} = \{T_{OMf1}, T_{OMf2},..., T_{OMfn}\}$ is a set of transitions that represent the output membership functions. T_{OMf1}, T_{OMf2}, T_{OMf3}, and T_{OMf4} represent the calculation performed by the used fuzzy method to compute the membership degree of respectively, $u_{large\_increase}(TR)$, $u_{increase}(TR)$, $u_{decrease}(TR)$, $u_{large\_decrease}(TR)$,
- $P_{OMd} = \{P_{OMd1}, P_{OMd2}, ..., P_{OMdn}\}$ is a set of places that represent the output membership degree. The places P_{OMd1}, P_{OMd2}, P_{OMd3}, and P_{OMd4} indicate that the output parameters of "AI is increased", "AI is increased largely", "AI is decreased", and "AI is decreased largely" are satisfied with the following membership degree, $u_{large\_increase}(TR)$, $u_{increase}(TR)$, $u_{decrease}(TR)$, $u_{large\_decrease}(TR)$, respectively.

IV. SIMULATION

The simulation of the proposed QoS architecture is studied with ns-2 simulator. Each mobile host has a transmission range of 250 meters and shares an 11 Mbps radio channel with its neighboring nodes. We compare the performance of FuzzyCCG with the 'original model' and SWAN model described in [4]. We use the word 'original model' to refer to IEEE 802.11 wireless networks without FuzzyCCG mechanisms. The simulation is realized in two steps: the first one investigates the performance of the proposed model in an environment characterized by a single shared channel. The second simulation considers a multihop environment.

A. Performance of a single shared channel

We consider a single hop environment that consists of a square shape of 150m x 150m. The simulation includes a variety of traffic types; FTP macro-flows, WEB micro-flows, and real-time flows. The video and voice flows representing real-time traffic are active and monitored for the duration of 100 seconds. Video traffic is modeled as 200 Kbps constant rate traffic with a packet size of 512 bytes. Voice traffic is modeled as 32 Kbps constant rate traffic with a packet size of 80 bytes.

The simulation considers a multiple scenarios of TCP best-effort traffic, 4 voice and 4 video flows. The TCP traffic is modeled as a mixture of FTP and Web traffic. Web traffic represents micro-flows, whereas FTP traffic corresponds to macro-flows. TCP flows are greedy FTP type of traffic with packet size of 512 bytes. Web traffic is modeled as short TCP file transfers with random file size and random silent period between transfers. The file size is driven from a Pareto distribution with a mean file size of 10 Kbytes and a shape parameter of 1.2. The length of the silent period between two transfers is also Pareto in distribution with the same shape parameter with a mean of 10 seconds.

We explore in Figs. 6 and 7, the impact of scalability of number of UDP video flows on the average end-to-end delay.

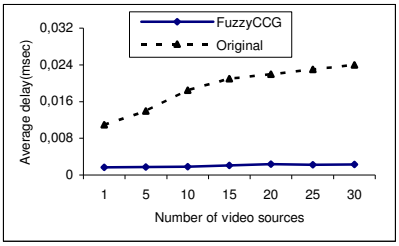


Fig. 6. Average delay in the original and FuzzyCCG models vs. number of video flows

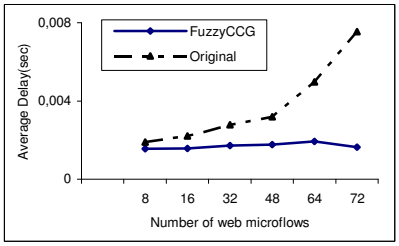


Fig. 8. Average delay in the original and FuzzyCCG models vs. number of web micro-flows

We consider a mixture of real-time traffic and TCP best-effort traffic which consists of 16 Web and FTP flows. It is observed in Fig. 6 that the original model shows an average delay larger than 12 msec with only 5 video flows and over 20 msec with 15 or more video flows. FuzzyCCG shows delays inferior to 2 msec with 5 video flows and less than 2.4 msec with 20 video flows. Hence, the reduction achieved by FuzzyCCG in terms of the average delay is about 90% in comparison to the original model. On the other hand, Fig. 7 illustrates that for up to 20 video flows, FuzzyCCG outperforms SWAN by about 13%.

Figs. 8 and 9 show the impact of the scalability of a growing number of web micro-flows on the average end-to-end delay. It is observed in Fig. 8 that the increasing number of web micro-flows has much more impact on the average delay in the original model than in FuzzyCCG. The average delay in FuzzyCCG remains around 1.8 msec, whereas in the original model the average end-to-end delay grows from 1.8 to 7 msec when the number of web micro-flows increases from 8 to 72 web micro-flows. On the hand, it is observed in Fig. 9 that the average delay in SWAN and FuzzyCCG models is similar for up to 16 web micro-flows. For the highest number web micro-flows, the average delay of traffic in FuzzyCCG becomes smaller than in SWAN by about 18%.

B. Performance in multihop environment

In what follows, the simulation considers a multihop network of 50 mobile nodes. The network area has a rectangular shape of 1500m x 300m. The AODV protocol [28] is chosen as a routing protocol. The flows traverse three intermediate nodes on average between source and destination. In this multihop network, we consider a mixture of real-time and TCP best-effort traffic. The real-time traffic is modeled as 4 voice and 4 video flows. The TCP traffic is modeled as a mixture of web micro-flows and FTP macro-flows traffic.

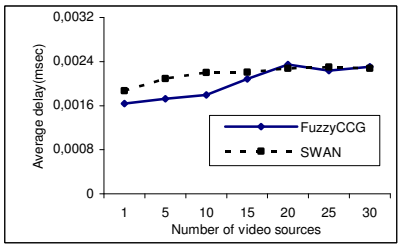


Fig. 7. Average delay in FuzzyCCG and SWAN models vs. number of video flows

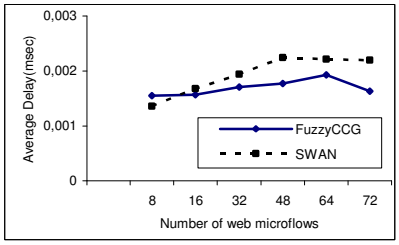


Fig. 9. Average delay in FuzzyCCG and SWAN models vs. number of web micro-flows

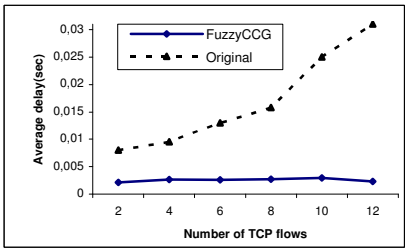


Fig. 10: Average delay in the original and FuzzyCCG models vs. number of TCP flows

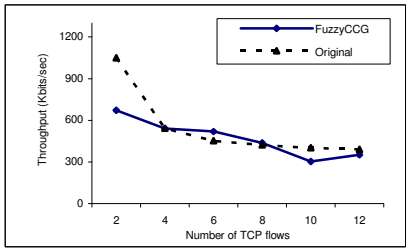


Fig. 11: Average throughput in the original and FuzzyCCG models vs. number of TCP flows

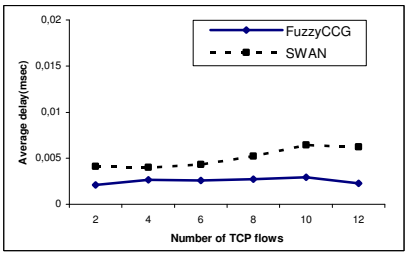


Fig. 12: Average delay in FuzzyCCG and SWAN models vs. number of TCP flows

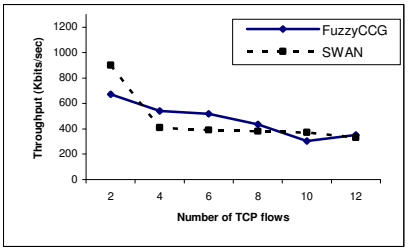


Fig. 13: Average throughput in FuzzyCCG and SWAN models vs. number of TCP flows

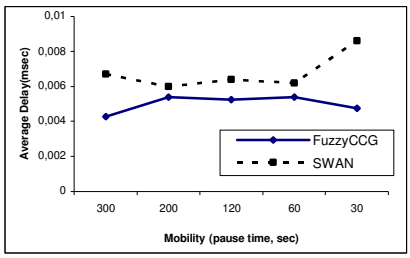


Fig. 14. Average delay in the original and FuzzyCCG models vs. mobility

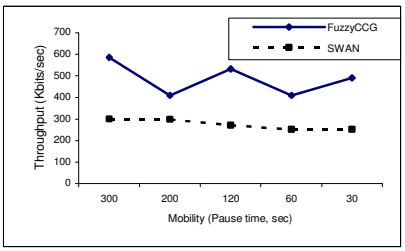


Fig. 15. Average throughput in the original and FuzzyCCG models vs. mobility

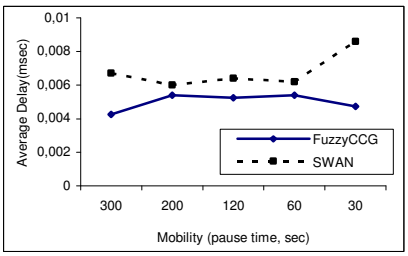


Fig. 16. Average delay in FuzzyCCG and SWAN models vs. mobility

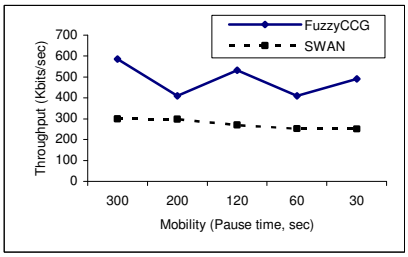


Fig. 17. Average throughput in FuzzyCCG and SWAN models vs. mobility

Figs. 10-13 explore the scalability impact of the increasing number of TCP flows on the average end-to-end delay and throughput of traffic. Fig. 10 illustrates a significant difference in terms of the average delay between FuzzyCCG and the original model. The average delay in FuzzyCCG grows slowly with the increasing number of TCP flows, and it remains between 2 and 3 msec. In contrast, the average delay in the original model grows from 7 to 31 msec as the number of TCP flows increases from 2 to 12 flows. Hence, the gain achieved by FuzzyCCG in terms of the average end-to-end delay, is by about 74-92%. Fig. 12 shows the average end-to-end delay in both FuzzyCCG and SWAN models. It is shown that the average delay is almost inferior to 3 msec in the proposed model, whereas in SWAN model the average delay is around 5 msec. This means that the achieved gain offered by FuzzyCCG is about 49% in terms of average delay.

Figs. 11 and 13 illustrate the impact of growing number of TCP flows on the average throughput of TCP traffic over the models of simulation. The average throughput of the TCP traffic in FuzzyCCG is almost the same as in the original model, as shown in Fig. 11. At lower number of TCP flows, the average throughput in the original model is superior to that in FuzzyCCG. A similar result is observed in Fig. 13 between FuzzyCCG and SWAN.

Figs. 14-17 explore the impact of mobility on the performances of FuzzyCCG. The real-time traffic is modeled in the same manner as discussed previously. The best-effort TCP flows consists of 5 web flows and 5 FTP flows. The

random waypoint mobility model is implemented at each node in the network. In the beginning, the nodes are randomly placed in the area. Then, each mobile node selects a random destination and moves with a random speed up to a maximum speed of 20m/s. After reaching the destination, the node will stay there for a given "pause time" then starts to move towards another destination. This process is repeated during all simulation time.

It is observed in Fig. 14 that the average end-to-end delay in FuzzyCCG increases slowly. The average delay in the proposed model remains almost less than 5.4 msec, whereas the average delay in the original model grows from 25 to 38 msec. This means that the proposed FuzzyCCG achieves a reduction in terms of average delay by about 79-87%. On the other hand, it is observed in Fig. 15 that the throughput of TCP best-effort traffic decreases slowly in the original model as the mobility increases. The average throughput in FuzzyCCG is superior to that of the original model by about 33% for different mobility scenarios.

Fig. 16 shows the average end-to-end delay with different mobility scenarios in both FuzzyCCG and SWAN models. For different mobility scenarios, the average delay offered by FuzzyCCG is about 10-36% better than that offered by SWAN. Fig. 17 shows that for different mobility scenarios, the throughput in FuzzyCCG is better than in SWAN model by about 43%.

The previous results show that the proposed architecture provides an average end-to-end delay with low and almost stable values, which is promising result for jitter-sensitive applications.

V. CONCLUSION

In this paper we proposed an intelligent solution for the congestion control of multimedia applications. The presented approach includes a fuzzy logic technique for buffer threshold management in order to show the ability of fuzzy thresholds to adapt to the dynamic conditions over the classical inflexible thresholds. In addition, the proposed solution includes a new technique based on fuzzy Petri nets to model and analyze the QoS decision making for buffer management in wireless ad hoc networks. It is observed in the simulation results that the proposed architecture can achieve a significant reduction in terms of the average end-to-end delay in comparison to both IEEE 802.11 and SWAN models. The obtained results confirm that the intelligent-based solutions can offer a good QoS support for multimedia services.

REFERENCES

[1] S. Chakrabati, A. Mishra, "QoS issues in ad hoc wireless networks", IEEE Commun. Magazine, vol. 39, N. 2, Feb. 2001, pp. 142-148.

[2] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", IETF RFC 2475, Dec. 1998.

[3] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview". IETF RFC 1363, June 1994.

[4] G.H. Ahn, A. T. Campbell, A. Veres, and L. H. Sun, "SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks," In the Proc. of IEEE INFOCOM, June 2002.

[5] S.-B. Lee, G.-S. Ahn, X. Zhang, and A.T. Campbell, "INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks," Journal of Parallel and Distributed Computing, special issue on wireless and mobile computing and communication, vol. 60, no. 4, Apr. 2000, pp. 374-406.

[6] H. Xiao, W. K.G. Seah, A. Lo, and K. Chaing, "Flexible QoS Model for Mobile Ad-hoc Networks," In the Proc. of IEEE Vehicular Technology Conference, vol. 1, pp 445-449, Tokyo, May 2000.

[7] Y. L. Morgan, T. Kunz, "PYLON: An architectural framework for ad-hoc QoS interconnectivity with access domains," HICSS'03 pres, Hawaii, USA, Jan. 2003.

[8] J.L. Sobrinho and A.S. Krishnakumar, "Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Networks," IEEE Journal on Selected Areas in Communication, vol. 17, no. 8, pp. 1353-1368, Aug. 1999.

[9] L. Khoukhi, S. Cherkaoui, "A Quality of Service Approach Based on Neural Networks for Mobile Ad hoc Networks," IEEE-IFIP Int. Conf. on Wireless and Optical Communications Networks, WOCN 2005, Mar. 2005.

[10] L. Khoukhi, S. Cherkaoui, "FuzzyMARS: A Fuzzy Logic Approach with Service Differentiation for Wireless Ad hoc Networks," In the Proc. of IEEE WirelessCom2005, Jun. 2005.

[11] C. R. Lin and J.-S. Liu. "QoS Routing in Ad Hoc Wireless Networks," IEEE Journal on Selected Areas in Communication, vol. 17, no. 8, 1999, 1426–1438.

[12] L. Khoukhi, S. Cherkaoui, "Flexible QoS Routing Protocol for Mobile Ad Hoc Networks," In the Proceedings of the 11th IEEE Int. Conf. Telecommunication (ICT2004), Brazil, Aug. 2004.

[13] S. Chen and K. Nahrstedt. "Distributed Quality-of-Service in Ad Hoc Networks," IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, 1999, pp.1426–1438.

[14] C.-R. Lin. "On-Demand QoS Routing in Multihop Mobile Networks," In Proc. of IEEE INFOCOM 2001, pp. 1735–1744, April 2001.

[15] S. Chen and K. Nahrstedt, "On finding multi-constrained paths," IEEE International Conference on Communication, 874 -879 June 98.

[16] C.C Lee, "Fuzzy logic in control systems: fuzzy logic controller- part I and II," IEEE trans. on Systems, Man, and Cybernetics, vol. 20, no. 2, 1990, pp. 404-418.

[17] L. A. Zadeh, "fuzzy logic = computing with words", IEEE Trans. on fuzzy systems," vol. 4, no. 2, 1996, pp. 104-111.

[18] A. R. Bonde and S. Ghosh, "A comparative study of fuzzy versus Fixed thresholds for robust queue management in cell-switching networks," IEEE Trans. Networking, vol. 2, no. 4, Aug. 1994, pp. 337–344.

[19] M. B., Dwyer, and L. A., Clarke, "A compact Petri net representation and its implication for analysis," IEEE Trans. Software Engineering, 22, 1996, pp. 794-811.

[20] T., Murata, T., Suzuki, and S. M., Shatz, "Fuzzy-timing high-level Petri net model of a real-time network protocol," In the proc. of ITC-CSCC 96, Seoul, Korea, July 1996, pp. 1170-1173.

[21] S.-M., Chen, J.-S., Ke, and J.-F., Chang, "knowledge representation using fuzzy Petri nets," IEEE Trans. on Knowledge Data Engineering, 2, 1990, 311-319.

[22] G., Looney, "Fuzzy Petri nets for rule-based decision making," IEEE Trans. on System, Man, and Cybernetics, 18, 1998, pp. 178-183.

[23] T., Cao, A.C., Sanderson, "Variable reasoning and analysis about uncertainty with fuzzy Petri nets," In the proceeding of 14 th Int. conf. on application and theory of Petri nets, Troy, NY, August, pp. 126-175.

[24] S.I., Ashon, "Petri net models of fuzzy neural networks," IEEE Trans. on System, Man, and Cybernetics, 25, 1995, pp. 926-932.

[25] A., Chaudhury, D. C., Marinescu, and A., Whinston, "Net-based computational models of knowledge processing systems," IEEE Expert, 8, 1993, pp. 79-86.

[26] L. A., Zadeh, "Knowledge representation in fuzzy logic," IEEE Trans. knowledge Data Engineering, 1, 1989, pp. 89-100.

[27] L. Khoukhi, S. Cherkaoui, "A Fuzzy Petri Nets QoS Model for Wireless Ad hoc Networks, The 2005 IFIP Int. Conf. on Intelligence in Communication Systems (INTELLCOMM), October 2005.

[28] C.E. Perkins, E.M. Royer, "Ad-hoc on-demand distance vector routing," IEEE Workshop on Mobile Computing Systems and Applications, pp.90 –100. New Orleans, LA, Feb. 1999.

Experimental results on the support of TCP over 802.11b: an insight into fairness issues

Francesco Vacirca and Francesca Cuomo
Infocom Department
University of Rome "La Sapienza"
Via Eudossiana 18, 00184 Rome, Italy
Email: {vacirca,cuomo}@infocom.uniroma1.it

***Abstract*— Great attention has been dedicated, in the recent years, to the WLAN standards that are opening the market to the short range and high data rate wireless services in the local and hot spot areas. Technically speaking, the main strength of the most quoted standard, the IEEE 802.11, is the fully distributed nature of the access scheme, that provides cheap and easy-to-install components, able to operate in the unlicensed spectrum, still guaranteeing broadband capabilities. The aim of this paper is to deeply investigate traffic issues in 802.11b networks by emphasizing the interaction between WLAN link layer parameters or Access Point buffer provisioning with uplink/downlink TCP fairness. The novel aspect is that this investigation is fully made in an experimental environment. A great portion of flows that are exchanged in a WLAN are TCP-based (e.g. FTP flows). We prove, with real experiments, that TCP suffers of some inequalities that derive to unfair bandwidth sharing between uplink and downlink. Our extensive experimental analysis shows the main effects of these inequalities on the TCP behavior and highlights some performance anomalies that are difficult to be measured via simulations.**

I. Introduction

Wireless LAN standards are drawing the attention of the research and industrial community due to their potentialities in opening the market to the short range and high data rate wireless services in the local and hot spot areas. Technically speaking, the main strength of the most quoted standard, the IEEE 802.11, is the fully distributed nature of the access scheme, that provides cheap and easy-to-install components, able to operate in the unlicensed spectrum, still guaranteeing broadband capabilities.

Several works regarding 802.11 WLANs have been published: analytic models (e.g. [1]), simulation environments (e.g. [2] and [3]), experimental works (e.g. [3], [4], [5], [6] and [7]). In this paper we deal with experimental evaluation of 802.11b and we specifically point out the unfairness issue when uplink and downlink TCP flows compete in a WLAN.

The unfairness problem in a typical WLAN configuration made up of one Access Point (AP) and several mobile stations (STAs) has been highlighted by several works. Some papers stress the unfairness between uplink and downlink traffic and the disadvantages caused when the number of stations increases. The problem is mainly due to the fact that, while each station contends the medium to transmit its own traffic, the AP, with the same access mechanism, contends the medium to transmit the whole downlink traffic directed to the various STAs. To send the downlink traffic the AP relies on a unique MAC queue. The immediate conclusion is that, when the number of STAs increases, the downlink system performance decreases steeply because the AP transmission opportunities decrease inversely with the number of uplink competing flows. A proposal to reduce this drawback is given in [8] where authors operate at Logical Link Control (LLC) layer to solve the unfairness due to the 802.11b MAC mechanism. In the LLC AP a number of queues equal to the number of STAs is introduced; on the other hand, each STA is equipped with only one queue. A scheduling algorithm is then introduced to suitably pass the LLC frames to the MAC layer. In [8], to allow a fair share of the available bandwidth between uplink and downlink streams, AP MAC queue is provided with a lower contention window value than STAs' queues.
A controllable resource allocation method between uplink and downlink traffic flows has been proposed in [9]. This solution is based on measurements of the current load performed by the AP and on adapting some AP MAC parameters to control the fair sharing of bandwidth. The efficiency of the proposed method has been demonstrated by Markov analysis and computer simulations.

The unfairness between downlink and uplink becomes more critical when the flows exchanged in the WLAN are TCP-controlled. The combination of TCP mechanisms with an unfair bandwidth sharing increases the unbalancing between downlink and uplink flows giving rise to deep unfairness events. In [6] the TCP fairness over 802.11 is discussed by showing: i) the effect of the AP buffer size in an experimental test constituted by one mobile TCP sender and one mobile TCP receiver; ii) the up/down throughput ratio derived by carrying out an extensive simulation study. The main conclusions are that the buffer size in the AP plays a key role in the observed unfairness and that TCP throughput ratio between up/down could become very high ($\simeq 800$), thus giving rise to deep unfairness. Authors of [6] also propose a solution to alleviate the unfairness that is based on the manipulation of TCP advertised window. Simulative analysis of the proposed solution shows that a 1:1 ratio is maintained, resulting in fair allocation bandwidth. Two problems however exist: 1) the solution is not tailored to TCP flows with different round trip

This work has been partially supported by the PRIN TWELVE project founded by the Italian Research Ministry.

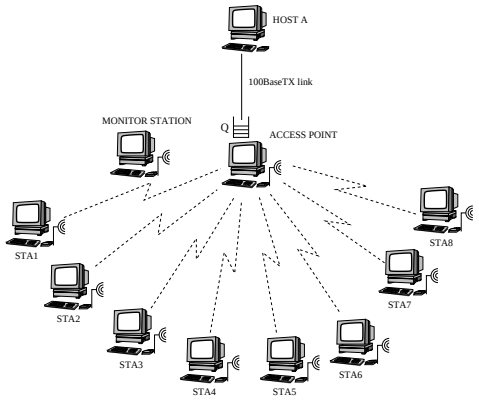


Fig. 1. Testbed layout.

times; 2) the advertised window manipulation requires that the AP is able to modify the TCP header fields and to re-compute the checksum; this could become time/resource consuming and results in a non-scalable solution.
The work in [10] considers unfairness at the TCP level due to different channel behaviors at the physical 802.11 layer. Authors propose an algorithm which improves the fairness among STAs that experience short channel failures.
Finally, several works suggest to exploit the upcoming standard IEEE 802.11e to solve the unfairness by differentiating the MAC access methods in uplink and downlink. The paper in [7] investigates the use of the 802.11e MAC EDCF to address transport layer unfairness in WLANs. A simple solution is developed that uses the 802.11e AIFS and CW_{min} parameters to ensure fairness between competing TCP uploads. Authors in [11] present measurements made using 802.11e wireless test-bed which shows how this new standard can be used to mitigate damaging cross-layer interactions between MAC and TCP. TCP ACK are prioritized by using suitable 802.11e MAC parameters in both the AP and the wireless STAs. This partially restores the fairness between uplink and downlink.

The aim of this paper is to deeply investigate the flow fairness in 802.11b by stressing the interaction between WLAN link layer parameters (e.g., ARQ retransmission persistence degree) and transport protocols. The novel aspect is that this investigation is fully made in an experimental environment constituted by an AP and up to 8 wireless STAs. Thanks to an extensive experimental analysis we are able to show the main effects of the 802.11b MAC on the TCP behavior and to propose a simple solution to alleviate the uplink/downlink unfairness.

The rest of the paper is organized as follows: Section II describes the testbed architecture and components. In Section III we report results of the measurement campaign focusing on the goodput behavior of downlink and uplink. In Section IV, we propose a simple mechanism, implemented in the AP, that mitigates the uplink/downlink unfairness. Main conclusions are drawn in Section V.

II. Description of the test-bed scenario

The test-bed reproduces a typical wired-cum-wireless scenario (Figure 1). It is composed of a 802.11b infrastructured WLAN and a dedicated 100BaseTX Ethernet link between the 802.11b AP and a PC (HOST A) that is the starting and/or the terminating point of all TCP connections. One PC acts as Access Point, 8 PCs as 802.11b client stations (from STA1 to STA8 in the figure), one PC equipped with a 802.11b network card monitors all the traffic on the air. The AP acts as a bridge between the wireless LAN and the 100BaseTX link exploiting the standard linux bridging functionalities [12]. In the test-bed topology, each STA is within the transmission range of all other STAs. All STAs are located in the same room and are motionless. The traffic is captured at:

- the monitor station, thus allowing the analysis of all the 802.11b frames exchanged on the air interface (this PC is equipped with a 802.11 wireless card that reads MAC headers and other 802.11b control information);
- the HOST A where it is easier to analyze the TCP evolution.

The adopted wireless LAN cards are 3COM 3CRDW696 802.11b driven by the Intersil Prism 2.5 chip-set [13]. All cards utilize the same firmware version (including the AP and the monitor station). The choice of this chip-set has been motivated by the high reconfigurability of the relevant options and by the possibility to use the HOSTAP driver [14] to implement a AP system on a linux PC. In fact, the Prism 2.5 chip-set (that is used both in WLAN host cards and in commercial APs) can be used to drive an AP in two modes: *Firmware-based* and *Host-based* AP mode. In the first mode, used in commercial APs, the chip-set utilizes a *tertiary* firmware for the AP functionalities such as authentication, association and forwarding of MAC frames. In the second mode, used in our testbed, the most time-critical actions are performed by the firmware (i.e. frame transmission, frame reception, beacon and probe frame handling), whereas other functionalities (such as authentication and association) are demanded to the host driver (the HOSTAP driver in our testbed). Moreover, the Prism MAC firmware implements a monitor mode that enables a 802.11b card to receive and pass to the host driver all frames with a PLCP header correctly received, irrespective of MAC frame check sequence errors, along with baseband layer information such as signal and noise levels.

The key MAC parameters (e.g., DIFS, SIFS, MAC header, etc.) are set according to the IEEE 802.11b standard. We set the MTU at 1500 bytes (fragmentation has been disabled) and the rate in the WLAN at 11Mbps. We disabled the RTS/CTS mechanism. Specific manufacturer features (out of 802.11b standard) have been disabled by default (e.g. power control, fallback rate control, etc.).
To study the effect of the AP buffer size on the uplink/downlink fairness, we exploit the standard linux traffic control tools [15] that enable us to modify the network interface card buffer size and queuing discipline.

The TCP version used in the experiments is the SACK-TCP [16] with window scaling and timestamp enabled [17]; TCP ACKs are sent according to the "delayed ACK" algorithm [18] (see [19] for a detailed insight into the linux TCP congestion control implementation). The TCP buffer sizes have been increased in order to avoid that the bottleneck of the TCP mechanism is the receiver advertised window. In this way we are able to capture all the effects of the congestion control interacting with 802.11 MAC access mechanisms.

Different software packages have been used in the test-bed: TCP traffic is generated to emulate bulk data transfer with a modified version of *ttcp* [20], changed to allow the *ttcp* server to accept multiple TCP connections simultaneously. The packet capturing tool is *tethereal* [21]. The traffic analysis and the performance metric computation have been performed with several *awk* [22] scripts.

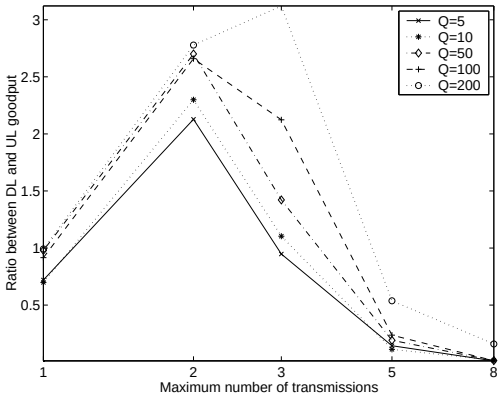
Experiments have been performed varying the maximum number of transmission attempts M_t at the 802.11b link layer, varying the AP buffer size Q and the scheduling discipline (i.e. FIFO and a custom scheduling discipline proposed to alleviate the TCP uplink/downlink unfairness problem). All the PCs are equipped with an additional 100BaseTX Ethernet card that it is used for control purposes. Experiments are configured and controlled by HOST A through *ssh* commands [23].
Each experiment lasts 500 seconds and all the metrics have been computed on the last 450 seconds of the experiments to remove the transient phase of TCP connections. A script runs at the end of every experiment to check consistency of the collected data and test-bed set-up: in particular, the number of active STAs and connections, the rate of all transmitted packets on the air and the absence of RTS/CTS packets are controlled[1]. The purpose of these checks is to enhance the test reliability.

III. Analysis of the experimental results

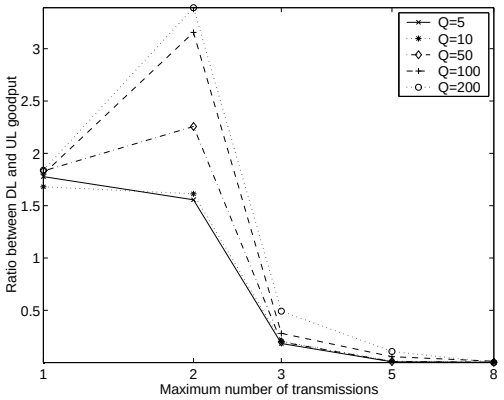
In order to understand the TCP uplink/downlink fairness issue in the 802.11b scenario, we focus our analysis on two main metrics:

1) the ratio between the overall TCP downlink goodput and overall TCP uplink goodput[2];
2) the packet loss probability estimated through the analysis of packet traces captured at the HOST A.

Two scenarios are considered: in the first one there are 2 TCP uplink connections (STA1-HOST A and STA2 -HOST A) and 2 downlink connections (HOST A-STA3 and HOST A-STA4). In the second scenario there are 4 TCP uplink connections (between STA1-4 and HOST A) and 4 downlink connections (between HOST A and STA5-7). In the remainder of this paper we refer to the first and the second scenario as dl2-ul2 and dl4-ul4 respectively. Given the symmetric characteristic of the scenarios, the ideal downlink/uplink goodput ratio is 1:1. We choose the symmetric scenario since it allows a comprehensive understanding of the balancing between uplink and downlink. In general, in typical WLAN scenarios, the symmetric assumption is not respected and the most of the traffic is in the downlink direction.



(a) Scenario dl2-ul2



(b) Scenario dl4-ul4

Fig. 2. Downlink/uplink goodput ratio vs. M_t, for different values of AP buffer sizes.

Figures 2(a) and 2(b) depict the downlink/uplink goodput ratio versus the maximum number of transmission attempts (M_t) at 802.11b link layer. The metric is measured for different values of the AP buffer size, with a FIFO scheduling discipline, in the dl2-ul2 and dl4-ul4 scenarios respectively. It is worth noticing that the goodput ratio is influenced significantly by both the maximum number of retransmission attempts and AP buffer size. As far as the behavior as a function of M_t is concerned, it can be noted that:

- when M_t=1, in case of dl2-ul2 scenario, the downlink/uplink goodput ratio is about 1:1 when the AP buffer size is large (between 50 and 200 packets), whereas the uplink is favored when the buffer size is smaller. In the dl4-ul4 case, downlink connections achieve a higher

[1]These checks are important in a scenario where it is difficult to distinguish between standard features and features developed by 802.11b card producers; e.g. by default the Intersil Prism 2.5 chipset decreases the transmission rate after a pre-defined number of unsuccessful transmission attempts.

[2]The goodput is defined as the throughput at TCP layer, excluding retransmitted packets.

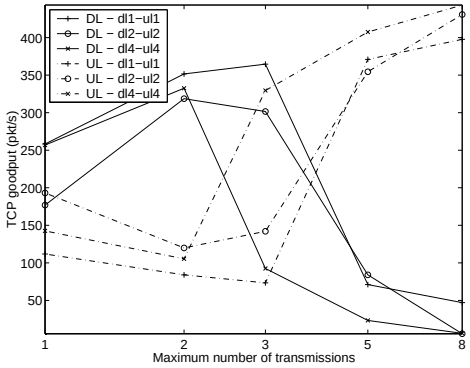
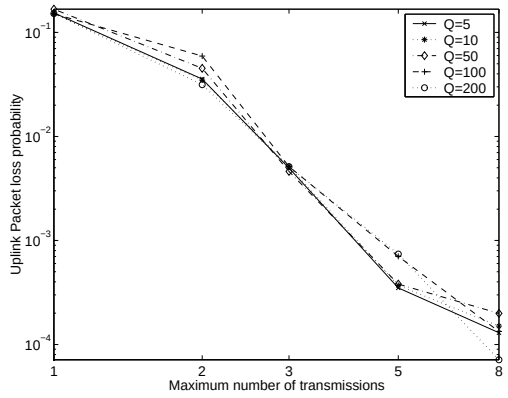


Fig. 3. Uplink and downlink goodputs vs. M_t, for the FIFO scheduling discipline (Q=100)

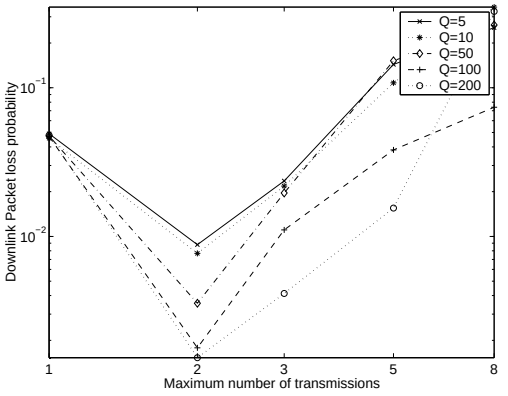
goodput with respect to the uplink connections;

- when the number of link layer retransmission increases (M_t=2) the downlink goodput increases in spite of uplink performance. The larger is the AP buffer, the more the downlink goodput is higher than uplink;
- by increasing M_t, the uplink connections seize the available bandwidth and the downlink connections starve. A larger buffer slightly alleviates the phenomenon.

Figure 3 depicts the downlink goodput (solid line) and uplink goodput (dot and dashed line) varying M_t, in dl1-ul1 (one TCP session in downlink and one TCP session in uplink), dl2-ul2 and dl4-ul4, when the AP buffer size is 100 packets.
Let us concentrate on the simple case dl1-ul1: for M_t=1, the downlink behaviour is satisfactory in terms of downlink/uplink goodput ratio. However, it could be noticed that the overall goodput is about 370 packets/s (250 packets/s in downlink and 120 packets/s in uplink). An useful expedient to improve the overall goodput in 802.11b is to increase the number of allowed transmission attempts at link layer to overcome the collision problem and minimize packet losses. The resulting performance anomaly is that, the overall goodput increases as expected, however, the downlink/uplink unfairness reverses for $M_t > 3$, favoring uplink connections in spite of downlink. This is mainly due to the different behaviour of TCP sender entities: TCP senders of uplink connections are directly connected to the bottleneck link, whereas in the downlink case the bottleneck is not in the access link. A better insight into these phenomena is given in Figure 4 where TCP DATA packet loss probability (estimated exploiting TCP packet retransmissions) is reported. Figures 4(a) and 4(b) depicts respectively the uplink and downlink packet loss probability in the dl4-ul4 scenario. We notice that, when M_t=1, the uplink packet loss probability is higher than downlink one, allowing downlink goodput to outperform uplink goodput. When the number of retransmission attempts increases, the uplink packet loss probability decreases monotonically, whereas the downlink packet loss probability decreases till M_t=2 and then it increases again. While the uplink packet loss probability is mainly due to packet collisions on the wireless channel, the downlink packet loss probability is the combination of two phenomena. On



(a)



(b)

Fig. 4. Uplink (a) and downlink (b) packet loss probability vs. M_t, for different values of AP buffer sizes.

the one hand there are packet losses due to collisions that decrease when M_t becomes larger. On the other hand, there are losses due to congestion in the AP buffer. These losses increase when M_t increases, because the congestion windows of TCP connections are able to inflate and fill the AP buffer. While uplink connections contribute to congestion the AP buffer with ACK packets, their performance is not influenced because DATA packets are not lost. It is to be considered that in traffic saturation conditions [1], every device achieves an equal portion of bandwidth (including the AP), leading to a ratio 1:(n+1) between downlink and uplink, where n is the number of STAs. With TCP as transport protocol, downlink flows are greatly influenced by congestion in the AP buffer and the ratio between downlink and uplink goodput decreases below the 1:(n+1) ratio.

Our experimental results confirm the influence of AP buffer size on the TCP uplink/downlink fairness problem (as also shown in [6]). With respect to [6], we show that the phenomenon is more complex and several factors influence the equilibrium between uplink and downlink connection goodput. We can summarize these factors as follows:

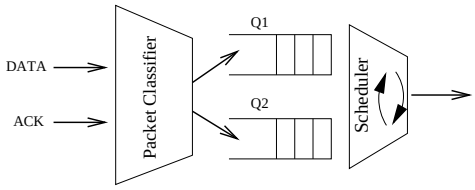


Fig. 5. Scheduling discipline.

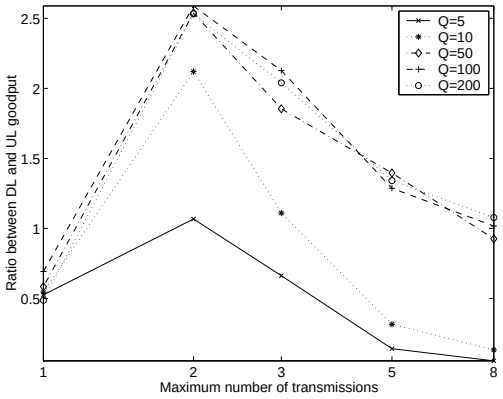
- Small AP buffer sizes favor uplink flows by increasing the downlink DATA packet loss probability;
- Large AP buffer sizes alleviate the throttling of downlink;
- A high number of transmission attempts favors the uplink;
- The downlink/uplink goodput ratio is unbalanced even in the dl1-ul1 scenario. The increasing number of supported TCP flows worsens the unbalancing phenomenon.

IV. TCP UPLINK-DOWNLINK SEPARATION VIA TRAFFIC CONTROL

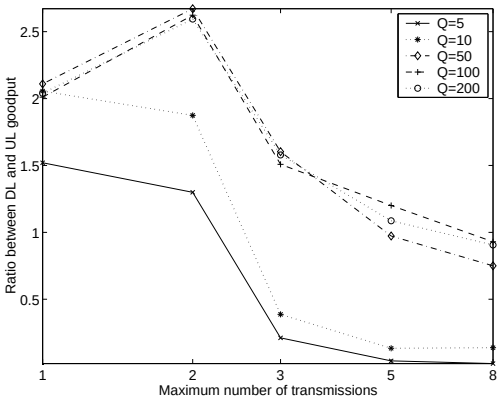
To increase the fairness between uplink and downlink in case of TCP flows we propose a simple traffic control mechanism. As well known, TCP sending rate is controlled by the rate the ACKs are received by the sender entity. The idea is to control the aggressiveness of the uplink flows by reducing the ACK rate issued by the AP. In this way the AP is able to control the throughput of the downlink versus the uplink one.

In the most of TCP implementations (see [19] and [24]), an ACK is generated at the TCP receiver side, every two DATA packets (according to the delayed ACK algorithm [18]). We implemented a simple scheduling discipline, at the AP buffer, that forces to 1:3 the ratio between ACK and DATA packets flowing through the AP towards the STAs. Since the reception of one ACK allows the transmission of two new packets, the uplink rate is forced to be the same of the downlink one because the ACK rate generates a doubled uplink data rate identical to the downlink data rate.

The scheduling scheme is represented in Figure 5. It is composed of a packet classifier that inspects packet characteristics (in our case TCP header fields) and forwards packets respecting user-defined rules to different queues. In case of our schedule, the classifier distinguishes between TCP DATA packets and TCP ACKs and enqueues them in Q1 and Q2 respectively. The scheduler is the entity that serves Q1 and Q2 in a weighted round robin fashion with a ratio of 2:3 for TCP DATA queue (Q1) and 1:3 for the ACK queue (Q2).
Figures 6(a) and 6(b) depict the ratio between the overall uplink goodput and the uplink one in the dl2-ul2 and dl4-ul4 scenarios respectively. Comparing Figure 2(a) with Figure 6(a) and Figure 2(b) with Figure 6(b), it is evident that in the region where downlink goodput is higher than uplink goodput the scheduler is not able to increase uplink/downlink fairness. When M_t increases and the AP buffer size is not too small (larger or equal to 50 packets), the scheduler is able to keep the goodput ratio about 1:1 indicating that uplink and downlink connections are experiencing the same goodput. Benefits of the proposed scheduler can be pinpointed by having a look



(a) Scenario dl2-ul2



(b) Scenario dl4-ul4

Fig. 6. Downlink/uplink goodput ratio vs. M_t, for different values of AP buffer sizes.

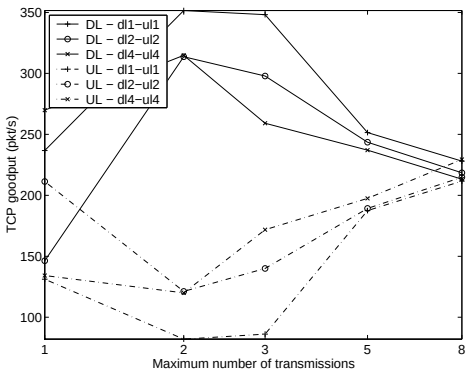
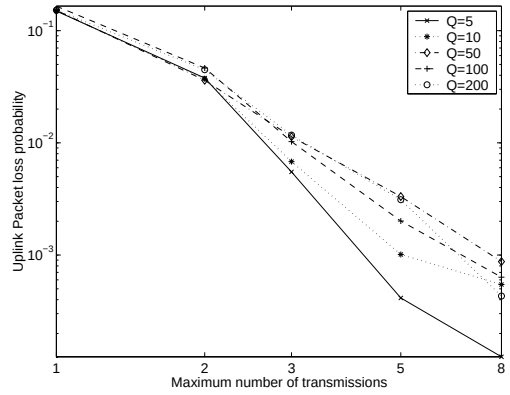


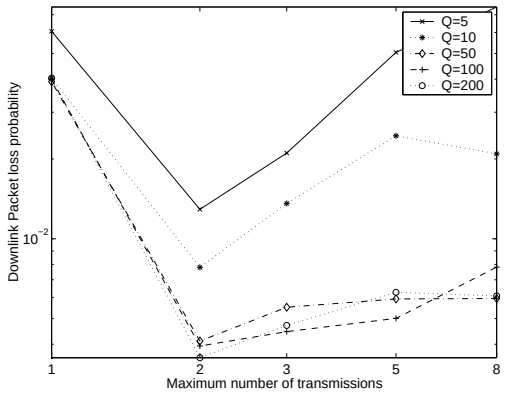
Fig. 7. Uplink and downlink goodputs vs. M_t, for the custom scheduling discipline (Q=100).

at Figure 7: in all the considered scenarios, the downlink and uplink goodputs converge to same value when M_t increases. It is to be noticed that already for $M_t > 5$ performance target is reached.

An insight into the packet loss probability experienced by TCP connections shows that the packet loss probability of



(a)



(b)

Fig. 8. Uplink (a) and downlink (b) packet loss probability vs. M_t, for different values of AP buffer sizes.

the uplink does not change with the customized scheduling discipline (comparison between Figure 4(a) with Figure 8(a)), whereas the downlink packet loss probability trend changes and the packet loss probability is reduced. The separation of the ACK scheduling in the AP presents two benefits: i) the TCP sender rate in the STAs decreases, ii) the ACK packet pressure in the AP MAC queue is reduced, diminishing downlink DATA packet losses.

It is worth noticing that our proposal is designed to equally share the 802.11b bandwidth between uplink and downlink not considering the number of active uplink and downlink connections. A more complex mechanism that estimates the number of active uplink and downlink connections and dynamically adapts the weights of the scheduler is needed to maintain the downlink/uplink goodput ratio proportional to the ratio between active downlink and uplink connections.

V. Conclusions

In this work, we deeply investigate the flow fairness in 802.11b in an experimental test-bed constituted by an AP and up to 8 wireless STAs. The aim is to highlight the interaction between WLAN link layer parameters and transport protocols. Thanks to an extensive experimental analysis we were able to show the main effects of the 802.11b MAC on the TCP behavior stressing in particular the effect of the AP buffer size, the number of link layer retransmission attempts and of concurring flows. The novelty of this contribution is a circumstantial report on TCP performance in a real 802.11b test-bed. To solve some performance anomalies we implemented in the AP a simple packet scheduling policy that succeeds in alleviating the uplink/downlink unfairness. The scheduling is a simple software module that can be included in the AP and that acts only by reading the TCP header fields of the exchanged TCP segments and by storing them in different queues handled with different priorities. It results easy to implement, scalable and not time consuming. Future work will be dedicated to adapt the proposed scheduling discipline to dynamic traffic conditions and asymmetric scenarios.

References

[1] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Function", IEEE Journal of Selected Areas in Telecommunications, Vol. 18, No. 3, pp. 535-547, March 2000.
[2] M. Methfessel et al., "Vertical Optimization of Data Transmission for Mobile Wireless Terminals," IEEE Wireless Communications Magazine, December 2002.
[3] M. Heusse, F. Rousseau, G. Berger-Sabbatel, A. Duda, "Performance Anomaly of 802.11b", Proc. of IEEE INFOCOM, 2003.
[4] G. Xylomenos, G. C. Polyzos, "TCP and UDP Performance over a Wireless LAN," Proc. of IEEE INFOCOM, 1999.
[5] A. Vasan and A. U. Shankar, "An Empirical Characterization of Instantaneous Throughput in 802.11b WLANs," technical report, available at http://www.cs.umd.edu/ shankar/selected-publications.html
[6] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP Fairness over Wireless LAN," Proc. of IEEE INFOCOM 2003, San Francisco, CA, USA, March 2003.
[7] D.J. Leith, P. Clifford, "Using the 802.11 e EDCF to Achieve TCP Upload Fairness over WLAN Links," Proc. IEEE WiOpt'05, pp. 109-118, April 2005.
[8] M. Bottigliengo, C. Casetti, C.-F. Chiasserini, M. Meo, "Smart Traffic Scheduling in 802.11 WLANs with Access Point," Proc. IEEE VTC 2003 Orlando, USA, October 2003.
[9] S.W. Kim, B.S. Kim, Y. Fang, "downlink and uplink Resource Allocation in IEEE 802.11 Wireless LANs," IEEE Transactions on Vehicular Technology, Vol. 54, No.1, pp. 320-327, January 2005.
[10] M. Bottigliengo, C. Casetti, C. F. Chiasserini, M. Meo, "Short-term Fairness for TCP Flows in 802.11b WLANs", Proc. of IEEE INFOCOM 2004.
[11] A.C.H. Ng, D. Malone and D. J. Leith, "Experimental evaluation of TCP performance and fairness in an 802.11e test-bed," Proc. ACM SIGCOMM Workshop on Experimental Approaches To Wireless Network Design and Analysis, Philadelphia, USA, August 2005.
[12] "Linux Bridging HOWTO," http://bridge.sourceforge.net/howto.html
[13] "RM025 - Prism Driver Programmers Manual," June 2002, Intersil.
[14] hostap, software available at http://hostap.epitest.fi .
[15] B. Hubert and alii, "Linux Advanced Routing & Traffic Control HOWTO," http://lartc.org/howto
[16] J. Mahdavi et al., "TCP Selective Acknowledgment Options", RFC2018, IETF, October 1996.
[17] V. Jacobson, R. Braden and D. Borman, "RFC 1323 - TCP Extensions for High Performance," IETF May 1992.
[18] M. Allman, V. Paxson, W. Stevens, "RFC 2581 - TCP Congestion Control," IETF, April 1999.
[19] P. Sarolahti, A. Kuznetsov, "Congestion Control in Linux TCP," In Proc. of USENIX'02, June 2002.
[20] ttcp, software available at http://ftp.arl.mil/ mike/ttcp.html.
[21] ethereal 0.10.4 release, available at http://www.ethereal.com.
[22] www.gnu.org/software/gawk/gawk.html.
[23] http://www.openssh.com/.
[24] D. MacDonald, W.Barkley, "Microsoft Windows 2000 TCP/IP Implementation Details," available at http://www.microsoft.com.

Throughput Unfairness in TCP over WiFi

Vasileios P. Kemerlis, Eleftherios C. Stefanis, George Xylomenos and George C. Polyzos

Mobile Multimedia Laboratory
Department of Computer Science
Athens University of Economics and Business
Athens 104 34, Greece
vpk@cs.aueb.gr, leste@aueb.gr, xgeorge@aueb.gr and polyzos@aueb.gr

***Abstract*—This paper presents a measurement study of TCP performance at an operational WiFi deployment. After presenting the network topology and the tools used to generate and analyze traffic, we examine the throughput performance of competing TCP connections. We investigate how throughput is divided among the participating wireless hosts with respect to signal strength, traffic direction and use of the RTS/CTS mechanism. Our study shows that while competing clients with comparable signal strength are treated fairly, achieving similar throughput values, clients with lower signal strength are treated unfairly, relinquishing a larger share of the available bandwidth to clients with higher signal strength.**

I. INTRODUCTION

The IEEE 802.11b standard for *Wireless LANs* (WLANs), also known as *WiFi*, is becoming increasingly popular worldwide for providing wireless Internet access in University campuses and many other public areas. WiFi client devices (WLAN network interface cards) are now becoming standard equipment for mobile devices such as laptops, PDAs and advanced cell phones. Moreover, WiFi infrastructure devices (Access Points or APs) are increasingly used even in households, providing wireless coverage for home networks. This popularity, along with its easy and cheap deployment, indicates that WiFi technology will be an integral part of any Wireless on Demand System with aspirations to appeal to the masses. Such systems will have many users with diverse needs and/or access rights.

The WLAN environment is characterized by the existence of multiple wireless clients competing for a share of the bandwidth. *Quality of Service* (QoS) tools (such as [1]) can be used to allocate a fixed share of bandwidth to each client. However, in order to apply efficient QoS policies for these clients, it is useful to be able to estimate the throughput that a client is likely to achieve for a given amount of bandwidth.

The objective of this paper is to investigate how TCP throughout is affected by various network parameters by measuring the throughput values achieved by the clients. The parameters examined in this study are: i) signal strength and signal to noise ratio, determined by wireless card gain and topological factors, such as distance from the Internet *Access Point* (AP) and interfering obstacles, ii) RTS threshold value, which effectively enables or disables the RTS/CTS mechanism and iii) traffic direction, classified as either uplink (from client to AP) or downlink (from AP to client).

Our study is not concerned with improvements of TCP performance over WLANs. Measurements so oriented can be found in [2] and [3]. Our work only aims to discover the rules governing throughput behaviour in WiFi, so that they may be used for purposes such as client classification in WiFi specific QoS policies. We focus on the behaviour of TCP as it carries the majority of the traffic encountered on our campus WLAN, most other WiFi deployments and, of course, the Internet (e.g., see [4] and [5]); while UDP based measurements reveal more about the underlying network, it is TCP performance that most users experience. Our major conclusions are:

1. Throughput is unfairly distributed between competing TCP connections experiencing unequal signal strength, at least in a max-min sense of fairness. “Strong” clients dominate the wireless medium, forcing “weak” ones to drop their transmission rate below their fair share.
2. The RTS/CTS mechanism generally has a detrimental effect on maximum achieved throughput, but it may help to reduce the unfairness in some circumstances.

In the remainder of this paper we first describe the measurement testbed, including the hardware and software used and the procedures employed, and then proceed to examine single connection measurements, which serve to establish the base relationship between the throughput of a TCP connection and the parameters under study. We use these measurements as a baseline for our multiple connection measurements, where we investigate the achieved throughput per client when two clients are simultaneously transmitting or receiving TCP traffic.

II. TESTBED SETUP

A. Hardware setup

The measurements took place in the operational IEEE 802.11b WLAN deployed in the *Computer Science Lab* (CSLAB) on the 2nd floor of the main building of the *Athens University of Economics and Business* (AUEB). The WLAN infrastructure consists of a Cisco 1200 IEEE 802.11b AP directly connected to a Linux based router (*Zeus*), allowing wireless nodes to access the Internet. We employed two (mobile) clients for our experiments, *Ares* and *Apollo*; one or both of them are active in the WLAN during tests. In order to limit the number of parameters that could influence performance, both clients used the same type of wireless network interface card. We used the Zoom Air 4100 PCMCIA 802.11b wireless cards which are fully compatible over the air with the AP, support nominal bandwidths of 1, 2, 5.5 and 11 Mbps and are equipped with a 2.2dbi dipole external antenna and the Prism I chipset. We confirmed that the performance of both our test hosts was symmetric by exchanging Ares' and Apollo's roles during our tests; results were almost identical. Table I shows the names and characteristics of each host. Ares and Apollo are laptops, while Zeus is a desktop machine.

TABLE I
DESCRIPTION OF HARDWARE EMPLOYED

Name	Processor	Network Interface
Zeus	Pentium III 450Mhz	802.11b AP via IEEE 802.3 100BaseTX
Ares	Mobile Pentium 1700Mhz	PCMCIA 802.11b – Zoom AIR 4100
Apollo	Mobile Pentium 1600Mhz	PCMCIA 802.11b – Zoom AIR 4100

Since Zeus acts as router, it has two identical IEEE 802.3 100BaseTX interfaces, one for Internet connectivity and one for connectivity with the AP. We used the `mii-tool` [6] to downgrade the interface between the AP and the router to IEEE 802.3 10BaseT, as preliminary tests showed that the fast Ethernet interface negatively affected the performance of the wireless LAN. This was due to congestion arising at the AP queue when forwarding packets from the (100Mbps) wired LAN to the (11Mbps) wireless LAN. Therefore, in order to avoid packet drops at the AP queue, we forced the wired LAN to a speed comparable to that of the WLAN.

All wireless client interfaces were set to operate at 11Mbps. We disabled the 802.11b rate adaptation mechanism in order to avoid unpredictable effects and test the native performance of TCP at a specific data rate. It should be noted that the rate adaptation mechanism seems to have unfairness issues of its own [7,8]. Table II shows the network settings used during our measurements. For each experiment performed, all hosts used the same settings.

TABLE II
DESCRIPTION OF NETWORK SETTINGS

Parameter	Value
ESSID	AUEB
Channel	1 (2412 MHz)
RTS Threshold	200 bytes (ON) / 2346 bytes (OFF)
Fragmentation Threshold	2346 (OFF)
Radio Preamble	Short

B. Software setup

All hosts ran the Debian GNU/Linux operating system [9], kernel version 2.4.27, using the `orinoco_cs` WLAN drivers as kernel modules. Preliminary tests showed that the newer 2.6.x kernels face performance problems due to major changes in the networking subsystem; therefore, we reverted to a more stable kernel version. Linux was selected in order to enable the use of the many freely available measurement tools and to provide us with full control of all network parameters. All hosts were in multiuser mode during tests, but no user tasks were executing on Ares and Apollo. Zeus was running the appropriate Network Address Translation / Firewall rules.

For the measurements we used the `ttcp` benchmark tool [10], which sends a number of packets of a specified size to a receiver using TCP (or UDP), reporting at the end various transfer related metrics. We also used the `tcpdump` tool [11] to record detailed logs of all packets sent and received by the wireless interfaces during each test. Logs were later processed by `tcptrace` [12], an analysis tool that provides statistical and graphical analysis of TCP/IP traffic, correlating incoming and outgoing packets in order to compute performance statistics at both the IP and TCP layers. These statistics include estimated congestion window size, number of out-of-sequence and duplicated segments, throughput and RTT.

C. Measurement procedure

We present below measurements with the clients in two different locations, while the AP is fixed. Location A is close to the AP and location B is far from the AP. Table III shows the signal strength, noise level and physical distance of each location from the AP. Channel noise was nearly constant during all tests, indicating that no sources of interference were present, while the signal strength experienced some deviation from time to time, due to the movement of people in the room.

TABLE III
DESCRIPTION OF MEASUREMENT LOCATIONS

Location	Signal Strength	Noise Level	Distance
Location A	-2dBm	-96dBm	2m
Location B	-91dBm	-96dBm	30m

The measurements consisted of executing `ttcp` with appropriate parameters in order to send 15MB of data; each run was monitored with `tcpdump`. The *Maximum Segment Size* (MSS) used by TCP was 1460 bytes, that is, the maximum LAN packet size minus 40 bytes for headers. The main test parameters were transfer direction (uplink/downlink), RTS (On/Off), and the *signal to noise ratio* (SNR) defined by the location of the clients. A test script automating the above procedure repeated each test five times, allowing us to estimate the variance. The script recorded `ttcp` output and SNR levels at both endpoints of the transfer. The `tcpdump` output files were used with `tcptrace` to generate dynamic TCP behaviour data that were plotted using `xplot` [13]. Among the possible measurements produced by `tcptrace`, we present in this paper the following:

- *Throughput*: It is computed by dividing the actual bytes transferred by the transfer time. This was used to double-check the throughput estimation of `ttcp`.
- *Congestion window*: As there is no direct way to determine the TCP congestion window without instrumenting the TCP code at the sender, the amount of outstanding (unacknowledged) data was used to estimate the congestion window size. The `tcptrace` tool measures the maximum, minimum, average and weighted average values for outstanding connection data.

III. Measurement scenarios and analysis

A. Single connection measurements

In this section we present measurements involving a single TCP connection between Apollo and Zeus. After preliminary tests in many locations, we decided to present results from locations A and B, as these were representative of the results seen in other locations. For each location, we performed tests in both directions (uplink/downlink) for two different RTS threshold values: i) 2346 bytes, which is larger than the segment size of the sender, thus disabling the RTS/CTS mechanism, and, ii) 200 bytes, which is smaller than the segment size of the sender but larger than the size of a TCP acknowledgment segment, thus enabling RTS/CTS for data but not for TCP level acknowledgments.

The following observations can be made from these tests:

1. Throughput values are substantially lower than the available bandwidth (about 50% at best). Some explanations for this behaviour are given in [2], [3], but they are not within the scope of this paper.
2. Uplink and downlink throughput differ. The throughput achieved when Zeus was the TCP sender (downlink) was higher in all cases (Figure 1). This asymmetry was expected due to the differences between client and AP hardware. The AP's (Zeus) radio is more powerful (during the experiments we configured it at 100mW) and is also equipped with two 3dbi antennas working in diversity mode. Thus, we had better reception when Zeus was the sender, verified by the fact that the received SNR at Apollo was higher than it was at Zeus.
3. Enabling the RTS/CTS mechanism has a negative effect on achieved throughput (Figure 1). This was also expected since the RTS/CTS mechanism imposes overhead for each frame sent, by first exchanging an RTS/CTS frame pair, leading to decreased TCP throughput.
4. Lower signal strength leads to a greater frequency of changes in the estimated TCP congestion window. The result is a more "jagged" diagram for the estimated congestion window at location B (low signal strength), as opposed to a smoother congestion window diagram at location A (high signal strength) (Figure 2).

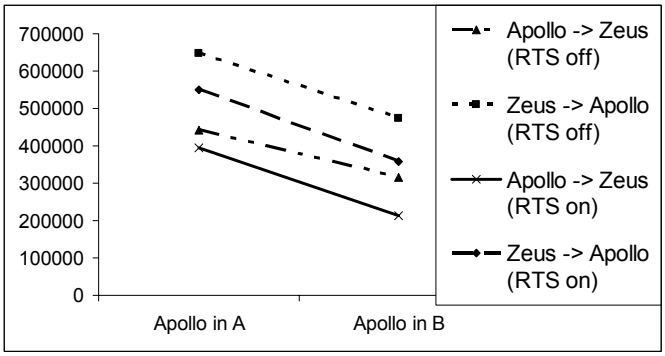


Fig. 1: Average uplink/downlink throughput (bytes/sec) (single TCP connection).

B. Multiple connection measurements

Having established above a performance baseline from the single connection measurements, we then added a second wireless client. Our goal was to investigate how the achieved TCP throughput was distributed between two competing TCP connections running on different hosts sharing the same wireless channel. We only used a single TCP connection per host since we were interested in the total throughput achieved by each host, not in the manner that throughput is distributed between competing TCP connections on the same host.

In the first scenario that we examined, the signal strength between the laptops and the AP was roughly the same and equal to its value in location A. This was achieved by arranging the hosts in a triangular topology where both clients were about the same distance from the AP and one another.

Again, we ran tests in both the uplink and downlink directions, with the RTS/CTS mechanism either enabled or disabled.

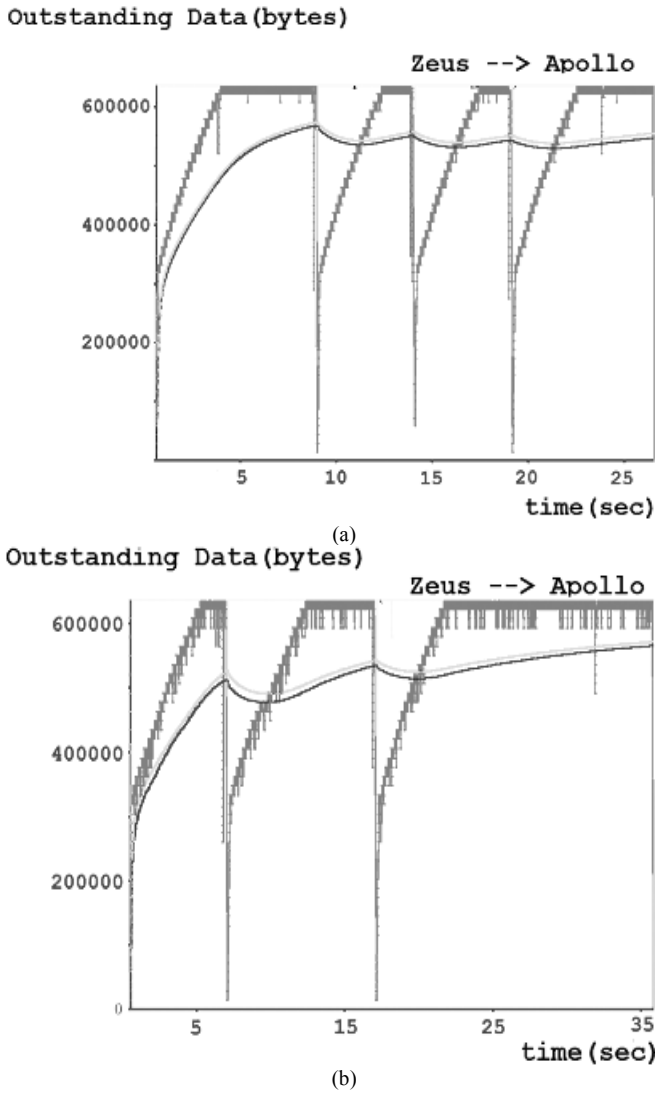


Fig. 2: Outstanding data (estimated congestion window size) at (a) location A and (b) location B for downlink traffic (RTS/CTS disabled, single TCP connection).

As the left side of Figure 3 shows, in this scenario total throughput was divided evenly among the two connections in both the uplink and downlink directions, whether RTS/CTS was enabled or not. This is particularly important in the uplink scenario, where the two clients are directly competing for use of the medium, unlike in the downlink scenario where the AP alone is sending data to both clients.

Another observation is that the total throughput achieved is greater with two competing clients than with one client. This is due to the conservative transmission behaviour of TCP: it is easier to get two TCP connections to send with rate R simultaneously than to achieve a rate of $2R$ with a single TCP connection. These results indicate that the IEEE 802.11 MAC protocol (CSMA/CA) is dividing throughput "fairly" when the signal quality is equal.

In addition, the observations that we made for the single connection measurements, that is, that an asymmetry exists between the uplink and downlink throughput and that the throughput is decreased when RTS/CTS is enabled, still apply. The RTS/CTS mechanism is unnecessary in this scenario, since the two clients can detect each other's transmissions.

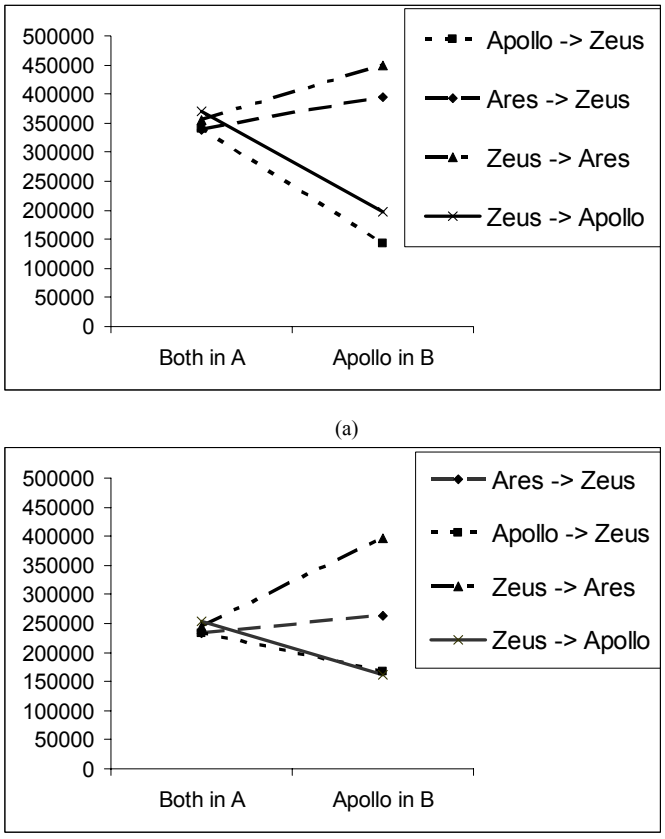


Fig. 3: Average uplink/downlink throughput (bytes/sec) with (a) RTS/CTS disabled and (b) RTS/CTS enabled (two TCP connections).

In order to investigate what happens when the competing clients experience different signal quality levels, in the second scenario Apollo was moved to location B, while Ares remained at location A, so that Apollo would experience worse signal level conditions than Ares. Throughput results from this scenario are shown on the right side of Figure 3.

In the downlink direction, with RTS/CTS disabled, in the single connection tests Apollo achieved a throughput of 474,680 bytes/sec at location B (Figure 1), while with Zeus

sending data to both Ares in location A and Apollo in location B, the throughput at location B decreased to 196,550 bytes/sec (Figure 3). Similarly, in the single connection tests Apollo achieved a throughput of 647,822 bytes/sec at location A (Figure 1), while with Zeus sending data to both Ares at location A and Apollo at location B, the throughput in location A decreased to 449,053 bytes/sec (Figure 3). We observe that the host at location A and the host at location B lost comparable amounts of bandwidth when competing against each other, as opposed to when operating in isolation: the throughput reduction in location A was 278,310 bytes/sec while at location B it was 198,769 bytes/sec.

The throughput reduction is however considerably different in relative terms: the host at location B lost 58.59% of its throughput, while the host at location A lost only 30.68% of its throughput. By comparing the two sides of Figure 3, we can see the reason: in the competing clients scenario the throughput of the host at location A (Ares) was considerably increased when its competitor was moved to location B (Apollo). That is, the TCP connection from Zeus to Ares took advantage of the decreased transmission rate of the connection from Zeus to Apollo, thus achieving higher throughput.

This unfair TCP throughput allocation is due to the delays incurred by MAC layer retransmissions of corrupted frames when Apollo is in location B, where the signal strength is lower and the frame error rate is higher. These delays are attributed to congestion by TCP, thus lowering the transmission rate of this connection and allowing the competing TCP connection to grasp a larger share of the bandwidth. As a result, the competing connection ends up with higher throughput than when both hosts had equal signal strength. This is evident by looking at the estimated congestion window sizes of Ares and Apollo in this scenario (Figure 4): Ares stabilizes its estimated congestion window at a large value, while Apollo keeps opening and closing it.

Enabling the RTS/CTS mechanism in the downlink direction decreased all throughput values, as expected. However, the overall behaviour followed the pattern observed with RTS/CTS disabled: when one host was moved to location B, the host at location A improved its performance at the expense of the host at B. This is reasonable, as in the downlink case the only data sender is Zeus, hence no conflicts occur and the sole impact of RTS/CTS is its overhead. In the reverse direction where there is contention, only acknowledgments are transmitted, which are too small to use RTS/CTS.

It was only in the uplink direction of this scenario that we expected the RTS/CTS mechanism to have some impact, since the two clients competing for the wireless medium were so positioned as to not be able to detect each other's transmissions all the time. We verified that this was the case by testing the link between Ares in location A and Apollo at location B and finding connectivity to be unstable. Indeed, in the uplink direction, with RTS/CTS enabled, the gap between the performance of the host at location A and the host at location B was considerably smaller (Figure 3). That is, the unfairness factor was smaller with RTS/CTS enabled, even though the total throughput achieved was worse: the ratio between Apollo's and Ares' throughput was 0.635 with RTS/CTS enabled, compared to 0.36 with RTS/CTS disabled. It seems then that in this case the RTS/CTS mechanism was beneficial with respect to the fairness of TCP throughput sharing between the two hosts at location A and location B.

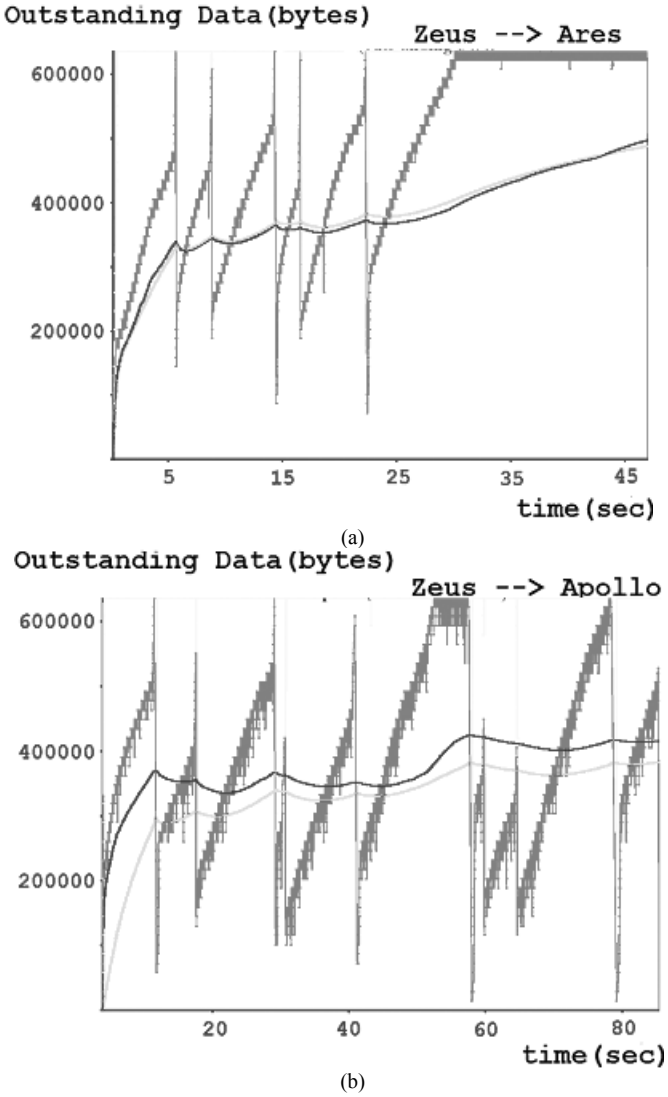


Fig. 4: Outstanding data (estimated window size) for downlink traffic (a) from Zeus to Ares and (b) from Zeus to Apollo (RTS/CTS disabled, two TCP connections, Ares in location A and Apollo in location B).

A possible explanation for this phenomenon is that the RTS/CTS mechanism, by introducing delays in the TCP transmissions and throttling back the TCP sender, makes the client at location A less aggressive when competing for throughput with the client at location B, thus preventing the client with the higher signal strength from grasping most of the available bandwidth. Apollo's throughput (Location B) may have remained low, but the decrease was smaller. Without RTS/CTS Apollo's throughput was reduced from 341,243 bytes/sec to 171,225 bytes/sec (49.82% decrease), while with RTS/CTS it was reduced from 232,567 bytes/sec to 166,615 bytes/sec (28.35% decrease). That said, the overall effect of enabling the RTS/CTS mechanism was clearly a negative one for both the individual and total throughputs achieved.

IV. Some Measurement Issues

Before we conclude our analysis, we discuss some measurement difficulties that we encountered and which we believe should be taken into account in similar tests.

1. Changes in signal strength levels can have great impact on TCP throughput. When the corridor between the AP and location B became crowded with people, for example, during class breaks, throughput dropped dramatically.
2. When performing measurements from locations with low signal levels, the results are more susceptible to random disturbances. In order to collect a valid set of results, such tests should be repeated a significant number of times.

V. Conclusions and Future Work

Our study indicates that there exist serious unfairness issues in the distribution of WiFi bandwidth among multiple hosts with different signal strengths when TCP is used. It is not trivial to predict how this affects the QoS policies applied in such an environment. As we observed in our experiments, when both clients were downloading data at the same time, the performance of the client with the lower signal quality was disproportionately affected by the client with the higher signal quality. By employing a QoS policy on the AP that enforces an upper bound on the rate of data sent to each client, for example, the throughput that each client would achieve if all clients had equal signal strengths, it is possible that clients with lower quality signals would not be penalized as much.

Self-organized wireless systems providing roaming between peers (e.g., see [14], [15]) may use QoS policies to assign bandwidth shares to users in proportion to their contribution. Our measurements indicate however that, despite nominally different bandwidth allocations, users with lower contributions may get better treatment due to their advantageous position. An issue for further study is how signal strength could be taken into account so as to retain the reciprocity motive.

Another issue for further study is the role of the RTS/CTS mechanism with respect to fairness. The scenarios that we studied show that this mechanism considerably reduces total TCP throughput achieved. On the other hand, measurements in scenarios where clients are greater in number and experience diverse signal quality levels indicate that the RTS/CTS mechanism can have a positive effect on the fairness of TCP throughput sharing. It is an open issue whether it is possible to obtain these benefits of RTS/CTS without its penalties.

Finally, we plan to extend our measurements to include:

1. Multirate tests in order to check the impact that the bit rate adaptation algorithm has on the transport layer.
2. UDP and/or more TCP implementation tests, in order to evaluate their performance with respect to fairness and correlate our results with the MAC layer.

Acknowledgment

We would like to thank Elias C. Efstathiou for his valuable assistance at various stages of this work.

References

[1] The tc tool, available at: http://developer.osdl.org/dev/iproute2/
[2] G. Xylomenos, G.C. Polyzos, P. Mahönen and M. Saaranen, "TCP Performance Issues over Wireless Links," *IEEE Communications Magazine*, vol. 39, no. 4, 2001, pp. 52-58.
[3] G. Xylomenos and G.C. Polyzos, "TCP & UDP Performance over a Wireless LAN," *IEEE INFOCOM 1999*, pp. 439-446.
[4] K. Thompson, G.J. Miller and R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics," *IEEE Network*, vol.11, no. 6, 1997.
[5] M. Fomenkov, K. Keys, D. Moore and K. Claffy, "Longitudinal study of Internet traffic in 1998-2003", *Winter International Symposium on Information and Communication Technologies (WISICT)*, January 2004.
[6] The mii-tool, available at: http://linux-ip.net/html/tools-mii-tool.html
[7] G.R. Cantieni, Q. Ni, C. Barakat and T. Turletti, "Performance Analysis under Finite Load and Improvements for Multirate 802.11," *Computer Communications*, vol. 28, no.10, pp. 1095-1109, 2005.
[8] M. Heusse, F. Rousseau, G. Berger-Sabbatel and A. Duda, "Performance Anomaly of 802.11b", *IEEE INFOCOM 2003*, March 2003.
[9] Debian GNU/Linux, available at: http://www.debian.org/
[10] The ttcp tool, available at: http://ftp.arl.mil/ftp/pub/ttcp/
[11] The tcpdump and libpcap tools, available at: http://www.tcpdump.org/
[12] The tcptrace tool, available at: http://jarok.cs.ohiou.edu/software/tcptrace/tcptrace.html
[13] The xplot tool, available at: http://www.xplot.org/
[14] E.C. Efstathiou and G.C. Polyzos, "Self-Organized Peering of Wireless LAN Hotspots," *European Transactions on Telecommunications*, vol. 16, no. 5, September/October 2005.
[15] P.A. Frangoudis, E.C. Efstathiou, and G.C. Polyzos, "Reducing Management Complexity through Pure Exchange Economies: A Prototype System for Next Generation Wireless/Mobile Network Operators," *12th Annual Workshop of the HP OpenView University Association (HP-OVUA)*, Porto, Portugal, July 2005.

Real-Time Multiplayer Game Support Using QoS Mechanisms in Mobile Ad Hoc Networks

Dirk Budke, Károly Farkas, Bernhard Plattner
Computer Engineering and Networks Laboratory (TIK)
Swiss Federal Institute of Technology Zurich (ETH Zurich)
Gloriastrasse 35, CH-8092 Zurich, Switzerland
Email: {budke, farkas, plattner}@tik.ee.ethz.ch

Oliver Wellnitz, Lars Wolf
Institute of Operating Systems and Computer Networks (IBR)
Technische Universität Braunschweig
Mühlenpfordtstr. 23, D-38106 Braunschweig, Germany
Email: {wellnitz, wolf}@ibr.cs.tu-bs.de

***Abstract*— Real-time applications, especially real-time multiplayer games, are getting popular in mobile ad hoc environments as mobile devices and wireless communication technologies are becoming ubiquitous. However, these applications have strict demands on the underlying network requiring low latency with a minimum of jitter and a small packet loss rate. Therefore, Quality of Service (QoS) support from the network is essential to meet the demands of real-time applications which is a challenging task in mobile ad hoc networks due to the high level of node mobility, properties of the wireless communication channel and the lack of central co-ordination.**

In this paper, we analyze and evaluate, via simulations, common QoS extensions, like backup route or priority queuing, and propose some new ones, such as real-time neighbor aware rate control policies or hop-constrained queuing timeouts, to ad hoc routing in IEEE 802.11 mobile ad hoc networks. We show how the performance, using these QoS extensions, of the selected AODV routing protocol improves significantly for connections up to three hops making possible to meet the demands of real-time applications. Moreover, we present how some common mechanisms like local repair of AODV and RTS/CTS degrade the performance and should not be used in case of real-time traffic.

I. Introduction

Online real-time applications such as multiplayer games have become very popular recently and are a big business in today's Internet expecting increasing revenue. Real-time applications share the common property that the information, for instance, the game character movements have to be delivered as fast as possible to the counterparts. If this data is delayed and arrives too late the game is not playable any more. Thus, real-time applications such as multiplayer computer games have strict demands on the underlying network and require low latency[1] connections, generally in the range of up to 50 – 150 ms, with a minimum of jitter and low packet loss rate below 5 % [1]. While multiplayer games are playable in the Internet, in wireless networks Quality of Service (QoS) mechanisms are required to meet the demands of real-time applications.

With the constantly increasing amount of powerful mobile devices, such as PDAs, laptops and mobile phones with wireless networking support, real-time applications and even multiplayer computer gaming in wireless environments are gaining much interest. Even though wireless networking infrastructure, such as base stations or access points, might not be available, various heterogeneous mobile devices can connect to each other setting up spontaneously a self-organized mobile wireless multihop ad hoc network (MANET) that does not rely on any existing infrastructure. However, ad hoc communication introduces several challenges mainly due to the mobility of the nodes, limited device resources, properties of the wireless channel and the lack of central co-ordination. The routing protocol, which is needed to send packets from a source to a destination via multiple hops, must cope with these challenges. Moreover, it has to be able to satisfy the QoS requirements of real-time applications, as well. But QoS provisioning in MANETs is even more difficult than in wired networks because of, among others, arbitrary mobility, weak wireless links, signal fading and interference, and the used channel access mechanisms. Thus, meeting the applications' demands in MANETs is very challenging and has been an open field of research in the last couple of years.

In this paper, we analyze and evaluate common QoS extensions, such as priority queuing, backup route, broken link detection, etc., and propose some new ones, like real-time neighbor aware rate control and hop-constrained queuing timeouts, to ad hoc routing in IEEE 802.11 [2] MANETs focusing on the requirements of real-time applications, specifically real-time multiplayer computer games. We selected the AODV [3] routing protocol as the starting point of our work because this protocol had shown the best performance in our initial simulations compared to some other ad hoc routing protocols, such as DSR, DSDV and OLSR [3]. However, our simulations show that end-to-end communication delay and jitter experienced using AODV are still too high to meet the demands of multiplayer computer games. In order to improve the performance of AODV for real-time applications, we require improved congestion handling mechanisms which can cope with mobility and the properties of an unstable wireless channel. Thus, in our work, following a cross-layer design, we have used: (1) QoS extensions to AODV like local repair and backup route; (2) traffic management mechanisms like priority queuing, timeouts, real-time neighbor aware rate control; and (3) MAC (Medium Access Control) layer support mechanisms like broken link detection, signal strength monitoring, neighbor

[1] We are using the terms *latency* and *delay* exchangeable in this paper

detection, RTS/CTS (Ready To Send/Clear To Send) adaptation.

We show in this paper, how the different QoS extensions do or do not improve the performance of real-time traffic in MANETs. First, we categorize and briefly discuss the extensions then present their combined impact on AODV's performance via simulations using the NS-2 network simulator [4]. Our initial routing protocol comparison results show that AODV outperforms the other routing protocols but still does not meet the demands of real-time applications. By applying priority queuing with timeouts and real-time neighbor aware rate control combined with broken link detection and backup routes, the end-to-end communication delay, delay jitter and loss rate can be reduced significantly. For connections up to three hops, AODV with the applied QoS mechanisms meets the demands of real-time multiplayer games. However, some common mechanisms like local repair of AODV and RTS/CTS degrade the performance of real-time traffic and should not be used.

The rest of the paper is organized as follows: In Section II, we discuss the issues of QoS provisioning for real-time multiplayer games. Then, we introduce the different QoS extensions in Section III and discuss them in Section IV. In Section V, we evaluate these extensions via simulations. We survey related approaches in the field of QoS provisioning in Section VI and conclude the paper in Section VII.

II. QoS Provisioning for Real-Time Multiplayer Games

Real-time applications, especially multiplayer games such as first person shooters, real-time strategy games, or sports games have strict QoS demands on the underlying network. Moreover, in MANETs, compared to wired networks like the Internet, these applications encounter additional problems due to the special challenges of MANETs. In this section, we discuss the substantial QoS requirements of real-time multiplayer games then describe our objective regarding to QoS provisioning in the light of the challenges of MANETs.

A. Requirements of Real-Time Multiplayer Games

In case of real-time multiplayer games, end-to-end communication delay, jitter and packet loss to a certain degree are the most important QoS attributes of the network, while the available network bandwidth is of less importance [5], [6]. For most real-time multiplayer games, a maximum of 150 ms round trip delay is still acceptable [1]. The effects of jitter, however, are not so well-researched yet. In [7], the authors suggest that latency and jitter are strongly coupled in the Internet with the ratio of jitter to latency being 0.2 or smaller. This also means that jitter in general is very small in the Internet if latency is below 150 ms and therefore has little impact on the game. Furthermore, players' perception of jitter is game-dependant [8] because high level of jitter usually leads to packets not arriving in time thus requiring the use of prediction mechanisms such as dead-reckoning. As the quality of these prediction mechanisms differ from game to game, players also perceive jitter differently. Furthermore, prediction mechanisms cannot always anticipate players' actions accurately so high level of jitter usually degrades the players' experience. Hence, the jitter level must be kept as low as possible. Moreover, the packet loss rate has similar effects and it should be kept in the range of 3 – 5 % for real-time multiplayer games [1].

B. Objective and Challenges

Our objective is to extend mobile ad hoc routing with QoS mechanisms by which the performance of the ad hoc routing protocol and the network can meet the demands of real-time multiplayer games. However, to give QoS guarantees in MANETs is a difficult task and the QoS extensions have to face the following challenges:

- **Mobility** - The topology of the ad hoc network might change unpredictably resulting in broken links and stale routes.
- **Congestion** - Real-time traffic must arrive in-time even if the network is highly loaded.
- **Shared Medium** - Wireless networks operating in ad hoc mode do not provide any QoS guarantees at the MAC layer due to the applied contention based medium access mechanism.
- **Wireless Signal** - The wireless signal suffers from fading and interference which gives rise to gray zones [9] and frequent retransmissions.

III. QoS Extensions to Mobile Ad Hoc Routing

In this section, we collect common and propose some new QoS extensions which are supposed to improve the performance of mobile ad hoc routing. Moreover, we classify these extensions into three different categories and point out the challenges they are supposed to tackle.

A. QoS Extensions

As QoS provisioning is a complex task and should be handled on several layers in the protocol stack we followed a cross-layer design. Thus, we classify the collected and proposed QoS extensions providing QoS support on the routing, interface queue and MAC layer, as illustrated in Fig. 1, into the following categories: (1) enhancements to the routing protocol; (2) traffic management; and (3) MAC layer support.

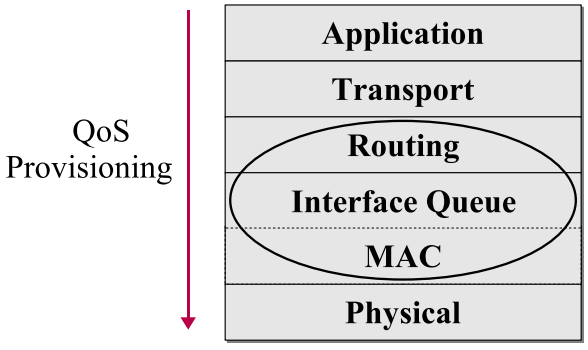


Fig. 1. Quality of Service Support in a Mobile Ad Hoc Network Node's Protocol Stack

TABLE I
QOS EXTENSIONS

		Challenges			
Category	**QoS Mechanism**	**Mobility**	**Congestion**	**Shared Medium**	**Signal**
AODV Enhancements	Local Repair	√			
	Backup Route	√			
Traffic Management	Priority Queuing		√		
	Timeouts		√		
	Rate Control		√	√	
Mac Layer Support	Broken Link Detection	√	√		
	Neighbor Detection	√	√		
	Signal Strength Monitoring				√
	RTS/CTS Adaptation			√	

Moreover, the different QoS extensions address different challenges of the mobile ad hoc environment. In Table I, we depicted the extensions and the corresponding challenges they are supposed to tackle.

1) AODV Enhancements: As the results of the ad hoc routing protocol comparison in Section V show, the reactive protocol *Ad Hoc On-Demand Distance-Vector* (AODV) [3] provides the best overall performance among the considered protocols. So, we selected the AODV-UU [10] implementation as the starting point of our work. To cope with mobility and broken links, AODV makes use of a *local repair* mechanism that allows intermediate nodes that have detected a link failure to temporally queue packets and repair the route. In addition, we extended AODV with the use of *backup route*s to repair broken links transparently and without any delays.

2) Traffic Management: When employing *priority queuing*, real-time packets are preferably transmitted to make sure that even in situations with higher load real-time packets do not become obsolete. Still, it might happen that real-time packets have to wait too long in the high priority queue. To prevent for the transmission of outdated packets we introduce hop-constrained queue *timeouts*. These are used to drop obsolete real-time packets and thus save bandwidth by applying a gradually decreased timeout timer. The actual timeout value at a given node is computed taking into account the number of already traversed nodes by the packet. In addition, to limit the amount of low priority traffic we introduce real-time neighbor aware *rate control* policies which prevent the occupation of the communication channel by nodes sending low priority traffic if other nodes have high priority traffic to be sent.

3) MAC Layer Support: Many features, already implemented in the IEEE 802.11 MAC layer, remain unused in higher layers and are implemented there again, however, less efficiently. With *broken link detection* based on Link Layer Feedback (LLF) the routing protocol is notified instantly if packets cannot be sent any longer over a certain link. In general, routing protocols periodically broadcast messages for *neighbor detection*. However, IEEE 802.11 ad hoc networks already broadcast periodic advertisements and thus the periodic routing messages are not required any more, neither for broken link detection nor for neighbor detection. With the help of *signal strength monitoring* more stable routes can be selected. In addition, IEEE 802.11 relies on the RTS/CTS (Ready To Send/Clear To Send) mechanism to avoid the hidden terminal problem. However, if used, *RTS/CTS adaptation* to the application's requirements is essential to not degrade the overall performance.

IV. DESIGN CONCEPTS

In this section, we describe our design concepts with regard to the mentioned QoS extensions and discuss these extensions in detail. We used NS-2 to implement these QoS extensions, so implementation details will be given where relevant. We did not implement and investigate all of these proposals in the simulator because, due to its limitations, we decided to do an implementation for a real hardware driver. Unfortunately, we could not investigate the real environment experiments due to time constraints.

A. AODV Enhancements

In order to extend the capabilities of the routing protocol to react efficiently to network topology changes and route failures we used *local repair* and enhanced AODV with *backup route*s.

1) Local Repair: AODV makes use of a local repair mechanism that allows intermediate nodes detecting link failures to queue packets temporally while it tries to repair the route. We used the built-in version of local repair from AODV.

2) Backup Route: AODV just stores the next hop entry to a certain destination in its routing table. If the link to that node breaks a new route discovery process has to be started which requires too much time for real-time data. The idea is to provide a backup route from the beginning that can be used instead. A backup route is a path with the same hop count as the default path but with a different next hop. For example, assume a scenario as illustrated in Fig. 2. Node S broadcasts RREQ messages to find a route to node D. Node I receives two RREQs, one from node 2 and the other from node 4. Both describe a 3-hop path to node S with the same sequence number. In this event, node I employs either node 2 or 4 as the next hop and uses the other as backup route. If a link to a neighbor on an active route breaks the backup route becomes the new active path. If hops with a higher hop count were used as backup paths, routing loops might occur (e.g., node I receives a RREQ from node 5 with a 5-hop path to node S

but this path contains a routing-loop and cannot be employed as backup route).

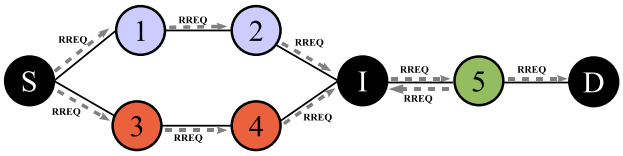


Fig. 2. AODV Extension Using Backup Routes

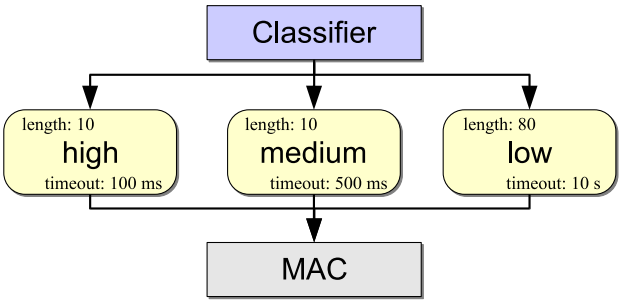


Fig. 3. Design of the Priority Queue

B. Traffic Management

The goal of traffic management is to differentiate among various types of traffic and give a higher level of service to high priority traffic at the cost of lower priority traffic. To achieve this, *priority queuing* can be used which we extended with hop-constrained queue *timeouts* and real-time neighbor aware *rate control* policies.

1) Priority Queueing: We used and modified the interface queue sublayer of NS-2 to implement priority queuing mechanisms. In our implementation, the interface queue consists of three sub-queues with different priorities and every packet is classified into one of three categories, as shown in Fig. 3, employing the TOS/DS field[2] in the IP header. All data packets that do not have any particular real-time QoS demands are marked as *low priority*, for instance, file transfer and e-mail traffic. AODV routing management packets such as RREQ, G-RREP[3] and RERR packets are marked as *medium priority*. And finally, real-time data packets are marked as *high priority* as well as AODV RREP messages if the high priority flag had been set in the corresponding RREQ. In contrast to medium and low priority, high priority packets are quickly outdated and dropped if they cannot be delivered in-time. Low priority packets might on the one hand experience higher delays, but on the other hand they are forwarded and delivered to the destination more reliably. Every sub-queue has a limited size and allows a packet only to be queued for a certain time interval. All sub-queues are exhaustive, meaning that a queue with lower priority will only be processed if all queues with higher priority are empty. In order to limit the amount of real-time and routing packets a node is allowed to transmit, the size of the high and medium priority sub-queues has been restricted to 10 slots. A much longer queue for real-time packets would result in more outdated packets. As the packets are marked with different priorities they can be handled specifically by the routing protocol and the interface queue.

The route request packets RREQ are marked with medium priority by default. By doing this, the actual current queue length of intermediate nodes are indirectly taken into account during the route discovery process in AODV. Nodes that have fewer packets in their high and medium priority sub-queues will be able to forward their packets quicker than nodes that already have to forward multiple high priority data streams. However, to minimize the delay in the event of broken links or the unavailability of routes for high priority data, we introduced an additional flag (bit 6 in the AODV RREQ header) that indicates that a RREQ is of high priority to enable nodes speeding up the route discovery process. But without marking the RREQ the network layer of the node that replies to the RREQ is not aware that the reply is urgent. Therefore, when the destination node receives a RREQ with a high priority flag, the TOS/DS field of the corresponding IP packet that holds the RREP message is set to high priority to make sure that the RREP is queued in the high priority sub-queue and forwarded as quickly as possible.

In addition to the interface queue, AODV itself queues data packets if no route to the destination is currently available. Because the route discovery process generally requires much more than 150 ms, we suggest that high priority packets are not to be queued by AODV but should be dropped immediately.

In order to cope with selfish nodes that simply mark all packets with high priority, packets could be filtered at the incoming interface to limit the amount of real-time packets from a certain node. However, we currently do not deal with malicious nodes in our implementation.

2) Timeouts: Real-time multiplayer games demand low latency connections with less than 150 ms round trip or 75 ms one-way delay, assuming symmetric latencies. However, packets that are delayed slightly more than 75 ms, might still be useful to prediction mechanisms such as dead-reckoning. Therefore, we decide to only drop packets which take more than 100 ms one-way delivery time and are clearly obsolete for real-time applications. Outdated real-time packets needlessly consume bandwidth and delay other real-time packets. Therefore, every packet that is stored in the queue is marked with a local time-stamp. When the packet is dequeued the time the packet spent in the queue is compared with the queue policy. The timeout interval for a high priority packet is 10 ms for every hop with a maximum value of 100 ms. Hence, the timeout mechanism supports only up to 10 hops from sender to destination which is fairly large for an ad hoc network. With every hop the packet passes, the timeout value is further decreased by 10 ms which approximates the

[2]Initially, Type Of Service now redefined as the Differentiated Services field

[3]In order to establish a bidirectional route between the source and destination nodes, an intermediate node that replies to an RREQ carrying the gratuitous flag sends an additional route reply message G-RREP to the destination

time it takes to process and forward the packet. The timeout information can be computed locally at each node by using the node's routing table because AODV, as a distance-vector routing protocol, knows about the number of hops to the source and the destination. The timeout mechanism is designed as a rather conservative approach and aims at dropping just the packets that will definitely come too late.

The medium priority sub-queue has a timeout of 500 ms which has shown reasonable performance in our simulations. Since both priority sub-queues are rather small and the packets are outdated quickly compared to the low priority sub-queue, additional mechanisms that prevent for starvation of low priority traffic have not been added.

3) Real-Time Neighbor Aware Rate Control: The current IEEE 802.11 protocol standards do not support QoS at the MAC layer and the medium access is carried out by the Distributed Coordination Function (DCF) handling every node equally [2]. All nodes have the same probability to gain access to the wireless channel and DCF does not distinguish between high priority and low priority data traffic. Thus, there is no guarantee that a node will send even high priority data packets within a certain time frame. Therefore, nodes that send high amounts of low priority data might consume most of the shared bandwidth. Due to the contention mechanism in DCF, other nodes which send high priority data might wait too long to access the channel and cannot transmit their high-priority data in time, although priority queuing has been used. The reason behind this is, that priority queuing works only locally and does not take neighboring nodes into account. To solve this problem at the MAC layer is one of the aims of the new QoS IEEE 802.11e [2] protocol that is still under development and not yet available. Even with the advent of IEEE 802.11e, the current standard will not be displaced immediately and thus, a mechanism is required that enables nodes that send real-time data to refrain other nodes from accessing the channel.

Our proposal is to limit the amount of low priority packets a node is allowed to transmit within a time interval depending on the amount of nodes handling real-time traffic in its neighborhood. Therefore, the interface queue needs access to the actual number of neighbors sending high priority traffic to adapt the rate control system of low priority traffic.

For this purpose, every node maintains a time-stamp *PRIO_TIMEOUT* that is updated when the node transmits a high-priority packet. If the node has sent real-time traffic within the last *PRIO_TIMEOUT* seconds the node marks broadcast routing messages with a real-time flag. Employing the unused bit 5 in the AODV header of RREQ and RREP messages, a node indicates whether it currently transmits high priority traffic or not. In addition to that, the routing table entry has been extended by a time-stamp *LAST_PRIO*. Every time a node receives high priority data or a routing message with the high priority flag, the *LAST_PRIO* time-stamp for that particular routing table entry is updated. Now, the actual amount of neighbors sending high priority traffic with an up-to-date *LAST_PRIO* time-stamp can be derived from the neighbor list and the interface queue can adapt the rate control of low priority traffic upon the number of neighbors sending real-time traffic. By introducing artificial delays for low priority traffic, nodes with high priority traffic have a better chance of sending their data in time. If the mobility rate of the nodes is low and nodes with real-time traffic have not sent routing messages for a while, the actual number of neighbors with real-time traffic might be higher, however. In addition, selfish nodes might just mark all routing packets with the priority bit. But we assume normal and our policy conform behavior of the nodes.

C. MAC Layer Support

In order to have accurate access to the current network state, such as an up-to-date neighbor list and the wireless signal strength for each neighbor, support from the IEEE 802.11 MAC layer is required. We use *broken link detection*, *neighbor detection*, *signal strength monitoring* and *RTS/CTS adaptation* as supporting mechanisms from the MAC layer.

1) Broken Link Detection: For broken link detection we employed the Link Layer Feedback (LLF) mechanism from the AODV-UU implementation. If a packet cannot be transmitted successfully, that is the sender does not receive an ACK from the destination within a certain time interval, the packet is retransmitted. Excessive retransmissions, however, can be reported by the MAC layer using LLF and this information can be propagated to the routing protocol to deal with the broken link. This mechanism has two benefits. First, it reacts to broken links by usually one order of magnitude quicker than relying on HELLO messages. And second, HELLO messages used for broken link detection are not required any more which reduces the routing overhead.

2) Neighbor Detection: To make routing decisions or management actions in the ad hoc environment an up-to-date neighbor list of a node is required. However, to maintain this list usually each application implements its own neighbor discovery procedure using network probes or HELLO messages. This results in message overhead and a waste of the scarce bandwidth. Since this service is commonly demanded it makes sense to provide a common interface to upper layers accessing the neighbor list from the MAC layer. Unfortunately, NS-2 does not provide a way to access the neighbor list on the MAC layer, thus we did not investigate its effect but decided to implement this feature in a Linux hardware driver. In a real environment implementation we can rely on BEACON messages that are sent out regularly every 100 ms by the IEEE 802.11 MAC layer if operating in ad hoc mode.

3) Signal Strength Monitoring: To provide for link stability and avoid the gray zones mentioned in [9] the signal-to-noise ratio (SNR) is measured for routing messages. If the signal strength of a RREQ is not beyond a certain SNR threshold the request is not processed. Thus, other neighbors that have received the RREQ with a higher signal strength will forward the packet and create a more stable route. AODV employs an *expanding ring search* based on the TTL (Time To Live) field in the IP header for dissemination of RREQ messages. If a node receives a RREQ which has been sent with the maximal TTL,

it does not drop the RREQ even though the signal strength is below the threshold, since no better route is available. Due to the limitations of the employed propagation model in NS-2, we did not implement signal strength monitoring for NS-2 but did a real environment implementation.

4) RTS/CTS Adaptation: NS-2 supports a certain packet size threshold to indicate if RTS/CTS should be used. By default the threshold is 0 and RTS/CTS is always enabled. We disabled RTS/CTS for real-time packets. Without RTS/CTS the transmission channel is shared equally among the contenting nodes. Moreover, it requires less bandwidth and we assume that it reduces end-to-end delay and delay jitter.

V. EVALUATION

In this section, we evaluate the introduced QoS extensions via simulations and discuss the results in the light of the demands of real-time multiplayer games.

A. Simulation Settings

In our simulations, we have used NS-2 including wireless extensions developed at CMU [11]. Throughout the simulations, each mobile node shares a 2 Mbit/s radio channel with its neighboring nodes using the two-ray ground reflection propagation model [12] and the IEEE 802.11 MAC protocol. The simulation scenario consists of 20 nodes in an area of 650x650 m^2. The transmission range of each node is 250 m, which is a typical value for WLAN in a free area without any obstacles. The nodes that do not take part in the multiplayer game follow the Random Way Point (RWP) model [11] with a uniformly distributed speed between 0-3 m/s and a pause time of 180 seconds. Player nodes are assumed to move only slightly, within a distance of 0-15 meters, since it is rather difficult to move and play at the same time with today's available mobile gaming devices. We simulated typical multiplayer game traffic [5] between the player nodes as high priority bidirectional UDP data flows with a Constant Bit Rate (CBR) traffic of 20 packet/sec and a packet size of 64 bytes. Additionally, we simulated five bidirectional low priority data flows between any two random nodes as background traffic using the same traffic pattern as the game traffic. We simulated 10 game sessions with a duration of 600 seconds using different seed values and then averaged the results. The detailed simulation parameters are depicted in Table II.

TABLE II
PARAMETERS OF THE USED SIMULATION SCENARIO

Parameter	Value
Simulation Time	600 s
Nodes	20
Mobility Model	Adapted Random Way Point (RWP)
Area	650x650 m^2
Speed	0-3 m/s
Pause Time	180 s
Traffic Type	CBR with 20 packet/sec
Packet Size	64 bytes

To evaluate the performance of the investigated QoS mechanisms, we used the following metrics:

- **Latency:** the average time in 'ms' it takes to transmit a packet from the source to the destination.
- **Jitter:** it describes how much the packets vary in latency and is determined by calculating the standard deviation of the latency.
- **Loss rate:** the loss rate determines the amount of sent packets in relation to the amount of packets that have not been received successfully at the destination.

B. Ad Hoc Routing Protocol Comparison

We have run some initial simulations to see which ad hoc routing protocol has the best performance and can be used as the starting point of our work. We investigated four different protocols, namely Ad Hoc On-Demand Distance-Vector (AODV), Dynamic Source Routing (DSR), Destination-Sequenced Distance Vector (DSDV) and Optimized Link State Routing(OLSR) [3].

With regard to latency and jitter, AODV showed the best average performance while all the other protocols showed much higher latency and jitter in the range of some hundred milliseconds. DSR's performance was by far the worst in these comparisons. Concerning loss rate, the reactive protocols (AODV and DSR) showed the best performance with less than 10 % packet loss while the proactive protocols (DSDV and OLSR) produced significantly higher losses.

Based on our initial simulation results, we selected AODV as the routing protocol of choice for our work because AODV showed the lowest latency, jitter and packet loss rate.

C. Effects of the QoS Extensions

In the following, we show and discuss our simulation results investigating the effects of the QoS extensions. Fig. 4, 5, and 6 show the impact of one QoS mechanism at a time.

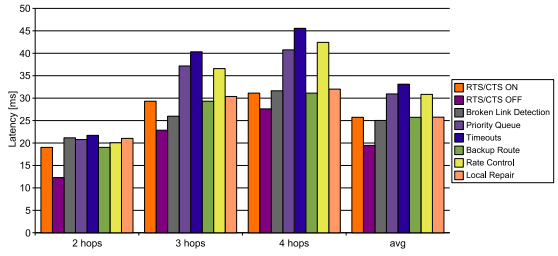


Fig. 4. Latency wrt Hop Count, and Average Latency of Game Traffic

From these figures we can see, that more or less all the QoS mechanisms, besides using RTS/CTS and local repair, improved somehow the experienced performance of the traffic, but it is hard to say clear statements about the level of the improvements. However, it is more interesting to see the combined impact of these QoS mechanisms. Thus, in the rest of the simulations we gradually extended the basic AODV protocol with the different QoS mechanisms in the order of the expected significance of the improvement caused by them and in the following figures (Fig. 7, 8, 9, 10, 11) every

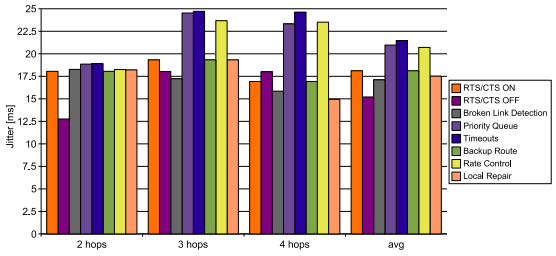


Fig. 5. Jitter wrt Hop Count, and Average Jitter of Game Traffic

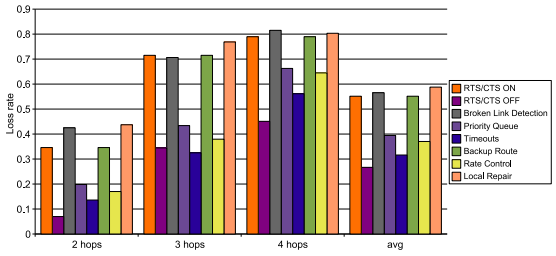


Fig. 6. Loss Rate wrt Hop Count, and Average Loss Rate of Game Traffic

bar represents the result applying its assigned plus all the preceding QoS extensions. For the evaluation we dropped all game packets with a delay of more than 100 ms because we assumed that they would not be of importance for the game anymore. With this strategy, we could reduce the average game traffic latency from approx. 300 ms to below 30 ms and the average jitter from above 50 ms to below 20 ms. Needless to say that as we dropped late packets on purpose we simply traded low latency for high packet loss. Fig. 7 and 8 show the latency and jitter experienced by the game traffic in case of different hop-number connections. As we can see from Fig. 9, we were able to reduce the game traffic loss rate significantly by introducing our QoS extensions. The average delay, jitter and packet loss rate for the background traffic are shown in Fig. 10 and 11.

Note that using local repair and RTS/CTS mechanisms degrade the overall performance in case of real-time traffic, thus we discuss them first and do not use them in the rest of the simulations. In the following, we will discuss the effects of every QoS extension in detail.

1) Local Repair: Using the local repair mechanism degrades the performance of the packet loss rate as well as latency and jitter. In particular, the average loss rate of high priority connections is increased by 5 %. First, this is due to the fact that the local repair process takes too much time for delivering real-time packets. Second, repaired routes tend to be longer than the original route and in this case the node sends an additional route error RERR message to the source which will itself restart the route discovery process again. Therefore, we do not recommend to use the built-in local repair mechanism of AODV in case of real-time traffic.

2) RTS/CTS Mechanism: The RTS/CTS mechanism degrades the performance of the routing protocol as well. The latency and jitter, and even the loss rate of the game traffic are significantly higher if RTS/CTS is enabled. The reason behind this is that multiplayer game data packets are very small in general and the RTS/CTS mechanism induces a huge overhead in this case decreasing the performance in all metrics significantly. As a result, RTS/CTS should not be activated by default and instead a threshold should be maintained by the system that depends on the number of neighbors and the size of the data packets that have to be transmitted. A neighbor list can be discovered easily by collecting MAC layer beacons from adjacent nodes.

3) Broken Link Detection: Using the Link Layer Feedback (LLF) mechanism of the MAC layer, broken links can be detected very quickly. The average latency can be reduced below 20 ms and jitter below 15 ms if LLF is activated. In addition, the packet loss rate can be reduced further by some percent as depicted in Fig. 9. Moreover, the amount of routing traffic is much smaller if LLF is used since AODV does not require HELLO messages for broken link detection any more.

4) Priority Queuing: To evaluate the effect of priority queuing, we handled the real-time game traffic as high priority traffic and the background traffic as low priority traffic. High priority packets that are not delivered within 100 ms (one-way) are regarded as lost and consequently the average latency and jitter of high priority packets are reduced significantly in all simulated scenarios, in our case to 20 ms and 14 ms. As a side effect, the packet loss rate increases as more packets are outdated. Regardless of the applied QoS mechanisms the loss rate highly depends on the number of hops. The higher the hop count, the more high priority packets are discarded since they arrive too late at the destination. However, when employing priority queuing, the loss rate of high priority packets can be substantially reduced as illustrated in Fig. 9. On the other hand, latency and jitter of low priority traffic is increased as Fig. 10 shows.

5) Timeouts: When adding timeouts to priority queuing, the loss rate of high priority traffic is slightly reduced further. Because of the small timeout value of the high priority queue, high priority packets which are late are now dropped earlier thus alleviating congestion. Connections with up to two or three hops have a loss rate of 2 % or 8 %, respectively. The average loss rate, however is still over 10 %. In addition, jitter and latency of low priority traffic can be also reduced.

6) Backup Route: The backup route mechanism increases the loss rate of high priority traffic for 2-hop connections slightly to 3 %. On the other hand, the loss rate of 3-hop connections is now reduced to 5 %. Using the backup route mechanism low priority traffic yields the lowest latency in all scenarios as depicted in Fig. 10.

7) Real-Time Neighbor Aware Rate Control: Applying our rate control mechanism the loss rate of high priority traffic has been reduced to 2 % and 4.5 % for 2-hop and 3-hop connections, respectively. Thus, 2- and 3-hop connections can be employed for real-time multiplayer games. Connections with 4 hops still suffer from a very high loss rate about 18 % and therefore cannot be used for real-time applications. Thus, the average loss rate of high priority traffic is just slightly

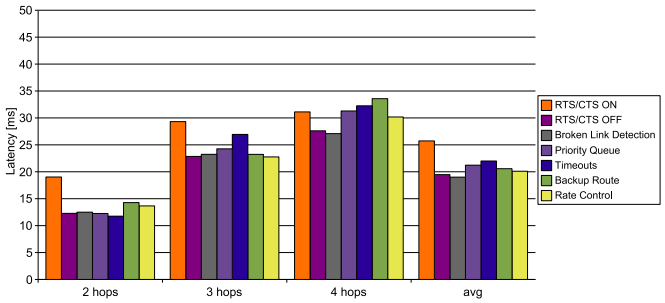


Fig. 7. Latency wrt Hop Count, and Average Latency of Game Traffic - Combined Impact

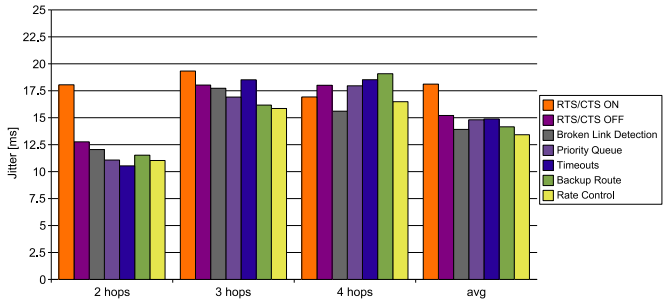


Fig. 8. Jitter wrt Hop Count, and Average Jitter of Game Traffic - Combined Impact

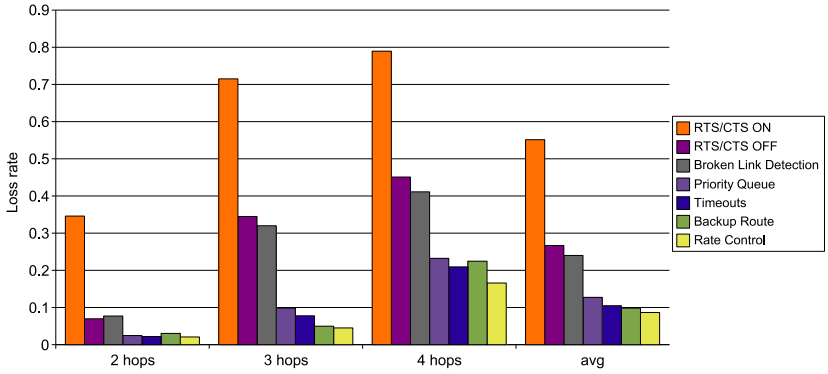


Fig. 9. Loss Rate wrt Hop Count, and Average Loss Rate of Game Traffic - Combined Impact

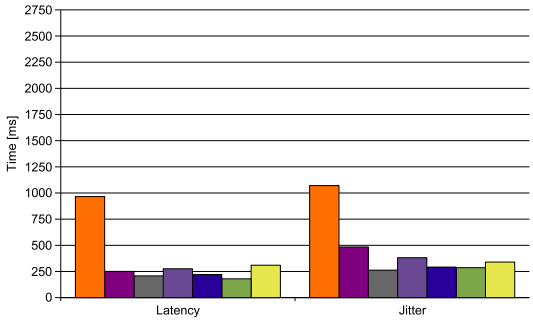


Fig. 10. Latency and Jitter of Background Traffic - Combined Impact

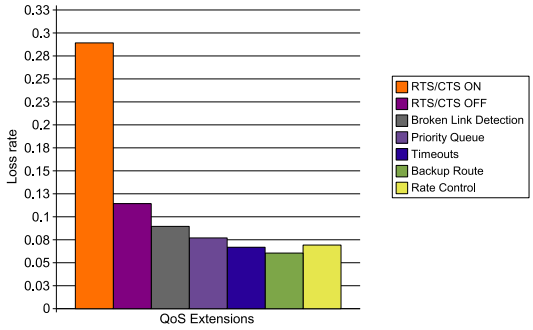


Fig. 11. Loss Rate of Background Traffic - Combined Impact

below 10 %. However, latency and jitter of low priority traffic are slightly increased since they remained in the same range in case of high priority traffic.

D. Summary of the Overall Impact of the QoS Extensions

We can see that by applying the discussed QoS extensions, except for AODV's local repair and the RTS/CTS mechanism, the packet loss rate for connections up to three hops can be reduced below 5 % since the end-to-end communication delay and delay jitter remain in the range of 25 – 30 ms and 15 – 20 ms, respectively. Moreover, the impact on the background traffic is also tolerable, i.e., the background traffic is not punished drastically in favor of the real-time traffic. Thus, other, non real-time applications can still use the network at the same time.

VI. Related Work

Many QoS architectures have been proposed for MANETs. Proposals like INSIGNIA [13], FQMM [14], or CEDAR [15] use a reservation-oriented approach and keep per-flow state information at the mobile nodes. Other approaches like SWAN [16] use a stateless feedback-based mechanism to achieve soft real-time services. All of these QoS architectures work at the network layer or above, so they make little use of information which is available at the lower layers. They also provide a self-contained solution to the QoS problem in MANETs. In this paper we take a look at individual QoS mechanisms to improve latency and jitter for real-time applications. While some of these mechanisms like priority queuing, rate control or backup route are also present in some of the QoS architectures mentioned above, mechanisms like broken link detection or

backup route belong to the routing part of the network layer and signal strength monitoring or RTS/CTS are features of the lower layers. We see this paper disjunctive to existing QoS architectures because most of the features presented here can also be applied to them.

One can find most of the QoS mechanisms mentioned in this paper also in related works.

The RTS/CTS mechanism is a common solution to the hidden/exposed terminal problem found in wireless networks and is part of the IEEE 802.11b standard [2]. As the overhead from RTS/CTS for small data packets is quite large, the standard also mentions a packet size threshold above which this mechanism should be used. Instead of just using the packet size, we proposed to also include the priority and the number of neighbors in the decision whether to enable RTS/CTS or not.

Broken link detection is part of the AODV routing protocol as defined in [17]. It also suggests that any suitable link layer notification, such as those provided by IEEE 802.11, can be used to determine network connectivity. Broken link detection with link layer feedback as well as signal strength monitoring is also used in the AODV implementation of Uppsala University [10] and is used unaltered in this paper.

In general, when looking at related work, most papers that discuss the performance of MANETs only deal with the analysis and the optimization of throughput and packet loss rather than taking latency and jitter into account. While on the other hand, papers on multiplayer games usually focus on the Internet as transport medium. In this paper, we apply existing and new QoS extensions to MANETs and compare their performance with the requirements of multiplayer games.

VII. Conclusions and Future Work

In this paper, we analyzed and evaluated common QoS extensions and proposed new approaches to mobile ad hoc networks. Following a cross-layer design, we classified these extensions into the three categories like routing protocol enhancements, traffic management mechanisms and MAC layer support mechanisms. In the simulations, we used the AODV routing protocol as the starting point of our work because AODV had shown the best initial performance compared to some other ad hoc routing protocols such as DSR, DSDV and OLSR. We have shown that, applying QoS extensions, the packet loss rate for connections up to three hops can be reduced below 5 % while the end-to-end communication delay and delay jitter remain in the range of 20 - 25 ms and 15 - 20 ms, respectively. This allows to meet the demands even of real-time multiplayer games. Moreover, we have shown how some common mechanisms like local repair of AODV and the RTS/CTS mechanism degrade the performance and should not be used in case of real-time traffic. However, as the network is getting bigger and the connections longer (consisting of more hops) giving QoS guarantees is extremely difficult without modifying the actual contention based medium access mechanism of IEEE 802.11 MAC layer.

As future work, we plan to investigate the discussed QoS extensions in a real test-bed environment. Moreover, we intend to design new mechanisms which can give protection against selfish nodes and which can provide quick channel access to nodes handling real-time traffic.

Acknowledgment

This work has been partly supported by the European Union under the E-Next NoE FP6-506869 project.

References

[1] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool, "The Effects of Loss and Latency on User Performance in Unreal Tournament 2003," in *Proceedings of the 3rd Workshop on Network and System Support for Games*, Aug. 2004, pp. 144–151.

[2] I.-S. S. Boards, "Part11-Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 12 June 2003, http://standards.ieee.org/getieee802/802.11.html.

[3] C. E. Perkins, Ed., *Ad Hoc Networking*, 2001.

[4] Information Sciences Institute ISI, "The Network Simulator NS-2," February 2005, http://www.isi.edu/nsnam/ns.

[5] Johannes Faerber, "Network Game Traffic Modelling," in *Proceedings of the 1st Workshop on Network and System Support for Games*, Apr. 2002, pp. 53–57.

[6] N. Sheldon, E. Girard, S. Borg, M. Claypool, and E. Agu, "The Effect of Latency on User Performance in Warcraft III," in *Proceedings of the 2nd Workshop on Network and System Support for Games*, May 2003.

[7] G. Armitage and L. Stewart, "Limitations of using real-world, public servers to estimate jitter tolerance of first person shooter games," in *Proc. ACM SIGCHI ACE2004*, 2004, pp. 257–262.

[8] M. Dick, O. Wellnitz, and L. Wolf, "Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games," in *Proceedings of the 4th Workshop on Network and System Support for Games*, Hawthorne, USA, Oct. 2005.

[9] H. Lundgren, E. Nordström, and C. Tschudin, "Coping with communication gray zones in 802.11b based ad hoc networks," in *Proceedings of the 5th ACM international workshop on Wireless mobile multimedia (WOWMOM'02)*. New York, NY, USA: ACM Press, September 2002, pp. 49–55.

[10] Erik Nordström, "AODV-UU implementation," http://core.it.uu.se/AdHoc/AodvUUImpl.

[11] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, October 1998, pp. 85–97.

[12] T. S. Rappaport, *Wireless Communications, Principles and Practice*. PRENTICE HALL INTERNATIONAL, 1996, ISBN: 0-13-042232-0.

[13] Xiaowei Zhang and Seoung-Bum Lee and Gahn-Seop Ahn and Andrew T. Campbell, "INSIGNIA: An IP-based Quality of Service Framework for Mobile Ad Hoc Networks," in *Journal of Parallel and Distributed Computing, Special Issue on Wireless and Mobile Computing and Communications*, vol. 60. Academic Press, Inc., Apr. 2000, pp. 374–406.

[14] Xiao Hannan and Chua Kee Chaing and Seah Khoon Guan Winston, *Quality of service models for ad hoc wireless networks*. CRC Press, Inc., Dec. 2002.

[15] Prasun Sinha and Raghupathy Sivakumar and Vaduvur Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm," in *Proc. IEEE INFOCOM 1999*, Mar. 1999, pp. 202–209.

[16] Gahng-Seop Ahn and Andrew T. Campbell and Andras Veres and Li-Hsiang Sun, "SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks," in *Proc. IEEE INFOCOM 2002*, June 2002, pp. 457–466.

[17] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," Nokia Research Center, University of California, University of Cincinnati, RFC 3561, July 2003.

Intensity-based Event Localization in Wireless Sensor Networks[1]

Markus Waelchli, Matthias Scheidegger and Torsten Braun
Institute of Computer Science and Applied Mathematics
University of Bern
{waelchli, mscheid, braun}@iam.unibe.ch

***Abstract*— Event detection and event localization are inherent tasks of many wireless sensor network applications. The inaccuracy of sensor measurements on the one hand and resource limitations on the other make efficient event localization a challenging problem. In this paper we propose a fully distributed localization scheme that consists of two algorithms. The distributed election-winner notification algorithm (DENA) performs the determination of the closest sensor node to an event and notifies all other nodes about that winner. The intensity-based localization algorithm (ILA) provides a signal independent position estimation of the event and is calculated at the winner node. The novelty of the ILA algorithm is its independence from the kind of signal emitted by an event. In contrast, it solely requires knowledge about the intensity of an event. The location of an event can thus be estimated without pre-knowledge about the nature of the event and with fewer constraints on the sensor hardware. These properties constitute the practicability of the algorithm in generic applications.**

I. Introduction

In sensor network applications event detection and localization are common features which imply two main challenges, namely how to observe a possible event location in a distributed manner and how to compute the location of an event efficiently and accurately. Due to battery and resource constraints on the sensor nodes some present approaches shift the computational burden of estimating the location of a node away from the sensor nodes to a sink node with more computational power and more memory. The main disadvantage of these approaches is the increased data traffic to provide the sink node with the necessary information to localize the event. The observation of an event on the other hand is commonly done by building clusters around predefined locations. Building up these clusters involves always communication overhead. This overhead is increased in wireless sensor networks, where battery constraint devices may follow sleep cycles to save battery power. Consequently, exchanging information may be expensive due to frequent topology changes or synchronization overhead. For these reasons, we intend to provide a fully distributed event detection framework that avoids the drawbacks of increased data traffic between the sensed area and the base station and does not need any cluster formation.

[1]The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

In our approach, the detection of an event (e.g fire burst) is observed as a set of derived values simultaneously sensed by a node (e.g. increased temperature, significant shockwave, etc.). Furthermore, the significance of the event can be determined by the sensor nodes, i.e. an event can be distinguished from background noise. This can be achieved by the use of fuzzy logics or probability theory. The task of filtering this background noise is not subject of this paper and will be considered in future work. Furthermore, the event is decreasingly observable the farther away a node is. We propose that considering these requirements all nodes in the relevant region can derive the intensity with which they sensed a certain event. Moreover, this intensity can be inferred as the barycenter of the set of deviated values sensed by the node. Thereby, each sensed value satisfies a certain membership function, e.g. 80° Celsius could have a membership degree of 0.8 in relation to the predicate 'hot'. A key idea of our approach is to use fuzzy logic mechanisms to classify the deviated values on the one hand and to infer the intensity of the event from these values on the other. The intensity of an event is consequently represented as a value in the interval [0, 1].

Using these derived values we propose to use a distributed election algorithm that determines the relevant subset of sensors, which are responsible for handling the event further, e.g. sending their information to a base station, or aggregate the information among each other. The determination of the relevant sensor nodes is performed fully distributed with a minimal overhead of information exchange.

II. Related Work

Event detection and localization are intrinsic features of wireless sensor networks. Much work in this context has already been done. The proposed schemes differ in the way they get range estimations and how they perform event observations. Some localization approaches ([1], [2]) depend on either a central instance such as a sink node or a cluster head, where the measurements from the sensors in the field are collected and the event location is computed. In [1] The distance of a sensor node to an event is approximated using the time of arrival (TOA) of the signal emitted by the event. The TOA values are routed together with the sensor node positions to a sink node, where the location of the event is computed as the maximum of a four-dimensional consistency function. [2] uses a cluster head approach to track the location of an event. Thus,

the overhead of sensor node to sink communication is avoided but additional communication to maintain the cluster structure is necessary. Another approach is Sextant proposed by [3]. Sextant uses Bézier regions to represent the locations of nodes as well as of events and does therefore without knowledge of the node positions. Additionally, Sextant is independent of a central instance. However, network properties which are needed by Sextant have to be disseminated in a restricted area, whenever one of them changes. Other approaches ([4], [5]) are mainly concerned in enabling and establishing group communication and data aggregation in a predefined area which has to be observed. Both algorithms require cluster formation what leads to extra communication overhead. A distributed algorithm for object tracking has been proposed by [6]. This approach supports event detection and tracking, but no event localization. A moving object is thereby tracked by a changeable cluster of nodes.

A common approach to estimate node locations is triangulation against known positions derived from reference points. APS [7] and GPS-Free [8] use angle of arrival (AOA) and time of arrival (TOA) respectively to calculate the position of a node. Both schemes are not practical for event localization as they depend on specific hardware. In contrast, we will propose a multilateration scheme that does only depend on the feasibility to sense an event on a sensor node.

III. Distributed Election-winner Notification

A key problem of event detection is the difficulty to identify and organize the sensor nodes, which are relevant for the event in a distributed manner with as little communication overhead as possible. To fulfill this task we propose the fully distributed election-winner notification algorithm (DENA). The DENA algorithm basically consists of two parts. In a first step the node closest to the event determines itself as winner node. In a second step the winner node notifies all other nodes about its election. The principle of the algorithm is depicted in Fig. 1.

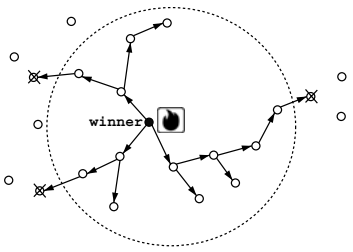


Fig. 1: Notification of election losers by winner node

The winner node broadcasts a notification message to inform all other nodes about its election. The notification message is thereby only retransmitted by sensor nodes that have overheard the event, i.e. the nodes bordered by the dotted line in Fig. 1. All other nodes having received the notification message (canceled in Fig. 1) simply ignore it. Additionally to its basic functionality of electing the winner node and distributing the notification message, the DENA algorithm offers functionality to provide the localization algorithm (see Section IV) with the information it needs. The DENA algorithm operates as follows:

1) Initially, each sensor node overhearing an event immediately calculates the intensity of the event as described in Section I. Furthermore, it schedules a notification message to inform all its neighbors about its election. The release time of the message is delayed according to the value of the derived intensity of the event, i.e. the higher the intensity the shorter the delay. We use the dynamic forwarding delay (DFD) mechanism proposed in [9]. A detailed description of the DFD concept we use is given below.
2) The sensor node closest to the event calculates the shortest delay. Consequently, it broadcasts the notification/IREQ message first. As it starts the notification message distribution it is implicitly signalized as election winner.
3) To gather the necessary information to perform the location estimation on the winner node, the notification message is combined with an information request message (IREQ) that has to be distributed within the two-hop neighborhood of the winner node. The reason for querying the two-hop neighborhood is given in Section V-B.
4) Each sensor node receiving the notification/IREQ message knows the existence of the winner node and immediately cancels its own election.
5) Each node that has received the notification message rebroadcasts it. Additionally, all nodes within two-hop neighborhood of the event generate an information respond message (IREP) to provide the winner with the position information it needs to calculate the event location. The IREP message may be piggy-backed on the notification messages of each two-hop neighbor to avoid additional transmissions.
6) The algorithm terminates when all election-losers have rebroadcast the notification message. All one-hop neighbor nodes of the winner perform their own location estimation as soon as they have overheard the piggy-backed IREP messages of their neighbors. Thereafter, they forward their results to the winner node which is responsible to calculate the final position estimation.

The intensity ω derived at each node in the reception area decreases with the distance d to the event. The general equation for this relation is $d \sim \sqrt[\alpha]{\frac{1}{\omega_{\min}}}$, where α is larger than one and $\omega_{\min}$ is the minimum intensity necessary to identify an event (see Section IV). The release time of the winner strongly depends on the amplitude of the event. Consequently, the weaker an event is, the slower it is detected by the DENA algorithm. Furthermore, it is crucial that each election-loser has to be notified before it determines itself as winner. This has to be taken into account for the design of the DFD function. The protocol proposed so far does not consider the case of collisions between transmissions. For this case we propose the usage of a backoff mechanism with an exponential time window after which the notification message is

rebroadcast if no notification message from another node was overheard in the meantime. We argue that collisions will not occur frequently, as the DFD is designed to avoid them. The simultaneous election of multiple winners is possible albeit not very probable. In this case, each winner node calculates its own position estimation and handles the result further, e.g. sends it to the base station. The algorithm does not yet consider any object tracking or the occurrence of simultaneous events. These are difficult topics and will be investigated in future work. Finally, an efficient broadcast protocol is used to minimize the number of retransmitting nodes. This protocol is again based on the DFD mechanism. The algorithm is discussed in the next section.

It must be mentioned that in our framework each sensor node knows its own location. This can be achieved by GPS, or by other location algorithms ([10], [11], [12]).

Dynamic Forwarding Delay

We use the dynamic forwarding delay (DFD) concept in two respects: Once to determine the release time of the notification/IREQ message, and once to perform an efficient broadcast in the reception area of an event. The DFD basically depends on the node position x and looks as follows:

$$DFD = MAX\_Delay \cdot f(x), \qquad f(x) \in [0,1]$$

The function f calculates a delay in dependence of the position of the message receiver. By the concept of the DFD, the decision to forward a packet is shifted from the sender to the receiver avoiding communication overhead to supply the sender with the information about its vicinity. This is in particular important in sensor networks, where battery constraint devices may follow sleep cycles to save battery power. In these networks gathering information about the neighborhood is expensive according to frequent topology changes or synchronization overhead. For this reason we think that a receiver based retransmission scheme is more appropriate. A key feature of the DFD mechanism used for the

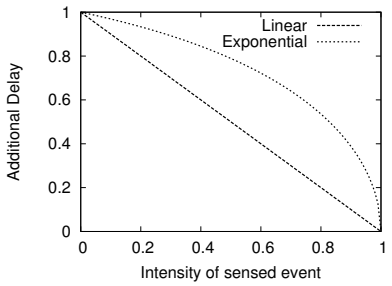


Fig. 2: DFD Functions

release time determination is the support of significant release time differences for nearby nodes. This is in particular true for the nodes close to the event. Thus, simultaneous winner election as well as collision probability are decreased. We propose an exponential DFD function as depicted in Fig. 2. For nodes with a higher significance, i.e. nodes that have derived a higher event intensity, the DFD timers are distributed over a larger interval. Thus, the probabilities of simultaneous winner election among these nodes as well as of collision occurrences are decreased. The exact specification of the DFD function for the notification/IREQ release has not yet been done. Its existence is however warranted and needs just the effort to find an adequate function and the according parameters. Thereby, the trade-off between efficiency and overhead caused by additional transmissions has to be minimized.

1) Dynamic Delayed Broadcast Protocol (DDB): Apart of using the DFD for the winner election, we use the DFD concept to perform an efficient broadcast. The algorithm operates in principle as follows: the DFD determines the time a node is delaying a broadcast message before rebroadcast it. While the expiration of this time, the node overhears the transmissions of other messages and cancels its rebroadcast if the retransmit threshold RT is under run, e.g. the distance to a sensor node that has already released the message falls below RT. The retransmission threshold RT may also be zero.

In [13] we have investigated different metrics for the rebroadcast decision. Thereby, the most efficient metric for the calculation of the DFD as well as for the decision whether to rebroadcast depends on the additional area a node may cover with its rebroadcast. Thereby, each sensor node calculates the additional area it covers as well as the DFD, whenever it overhears the transmission of the broadcast message. The node with the shortest DFD releases the packet first. The DDB protocol has been evaluated extensively and compared to well-known protocols. It was shown that the DDB protocol performs even better than a neighbor-based protocol in terms of energy consumption in most simulations. This is in particular true under frequent topology changes, what makes it useful for sensor networks where nodes often follow sleep cycles and the topology consequently changes frequently. Detailed information can be found in [13]. The main drawback of the DDB approach is however its computational complexity. To minimize this complexity, but nevertheless benefit from its advantages we redesigned the DFD metric. In the rest of this section we first shortly introduce the DDB concept with additional area coverage and then introduce our new metric which approximates the additional area approach, but uses much fewer energy.

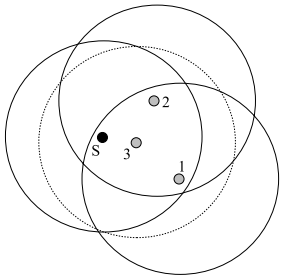


Fig. 3: Broadcasting with additional coverage

The basic functionality of broadcasting with additional area coverage is depicted in Fig. 3. The source node S starts

the communication by broadcasting its message. All three receivers calculate their DFD according to the additional area they cover with their retransmission. Node 1 is farthest away from S and accordingly calculates the largest additional area it would cover with its rebroadcast what leads to the shortest DFD. Nodes 2 and 3 overhear that retransmission and recalculate the additional area they newly cover and the respective DFD. Node 2 calculates the shorter DFD and rebroadcasts its message next. Node 3 overhears this message again and cancels its broadcast as it is totally covered by the transmissions of the other nodes. The DFD of the additional area coverage approach is calculated as follows:

$$DFD = MAX\_Delay \cdot \sqrt{\left(\frac{e - e^{\frac{AC}{AC_{MAX}}}}{e-1}\right)}$$

The additional area is denoted by AC and is always between zero and the maximal additional area AC_{MAX} a retransmitting node may cover. AC_{MAX} is achieved when a node is exactly placed on the border of the previous sender. In this case it covers an additional area of $\sim 61\%$. As mentioned above, the main drawbacks of this approach are its computational complexity and its memory demand. To reduce this overhead we redesigned the DDB by using a new metric. In Fig. 4 an approximation of calculating the additional area by the usage of the triangle connecting any three neighbor nodes is depicted.

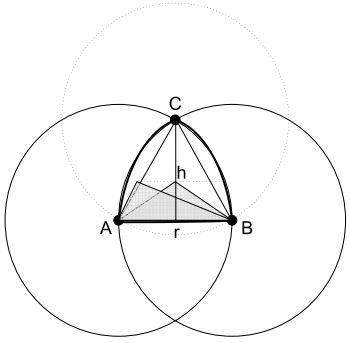


Fig. 4: Broadcasting with the triangle metric

Assuming that nodes A and B have already released their message, any node C that has overheard both transmissions lies in the intersection area of both transmissions. Moreover, the additional area a node covers is larger the farther away C from the connecting line $\overline{AB}$ is, largest at the position of node C in Fig. 4, i.e. the area A_{MAX} in this case is $\frac{\sqrt{3}}{4} \cdot r^2$. Consequently, we use the area of the triangle built by A, B and any node C in the intersection area as an indicator for the additional area a node C covers. Thereby, all nodes C with the same distance h to $\overline{AB}$ calculate the same triangular area, i.e. all nodes on the parallel line to $\overline{AB}$ with distance h. This adds a certain error, as the additional area a node covers depends on its location on the parallel line. The error is maximized at the center of line $\overline{AB}$. The area of the according triangle is zero, whereas the additional area a node covers is:

$$4\left[\int_0^{\frac{r}{4}} \sqrt{(r^2-x^2)}\mathrm{d}x - \int_{-\frac{r}{2}}^{-\frac{r}{4}} \sqrt{(r^2-x^2)}\mathrm{d}x\right] \sim 0.021r^2\pi$$

We argue that a maximal deviation of about 2% is tolerable and should not affect the algorithm in a destructive way.

Once determined, the area $A_{Triangle}$ of the triangle is used to calculate the DFD:

$$DFD = MAX\_Delay \cdot \sqrt{\left(\frac{e - e^{\frac{A_{Triangle}}{A_{MAX}}}}{e-1}\right)}$$

The reason to use an exponential function is to favor nodes with a bigger triangle and to minimize the probability of collisions among these nodes. Obviously, the node with the shortest DFD broadcasts first. With this broadcast a triangle as mentioned above is virtually created. All nodes located within this triangle cancel their retransmission of the message (a similar approach to cancel the retransmission was used by [14]). This test is very simple and can be easily calculated by the use of barycentric coordinates. It adds however some errors. The problem is depicted in Fig. 5:

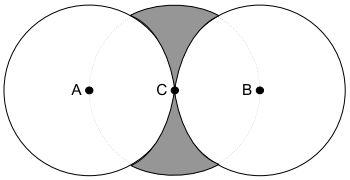


Fig. 5: The maximal loss of additional coverage

If the node overhearing two sending nodes is positioned exactly on the border of each node it must lie on the center of the connecting line between both nodes. In this case it cancels its retransmission, as it is within the triangle with height h = 0. The additional area it could cover is however $\sim 22\%$:

$$4\left[\int_0^{\frac{r}{2}} \sqrt{(r^2-x^2)}\mathrm{d}x - \int_{\frac{r}{2}}^{r} \sqrt{(r^2-x^2)}\mathrm{d}x\right] \sim 0.22r^2\pi$$

The determination of the DFD as explained in this section is only applied if a node has already overheard at least two messages. In the case of the broadcast initiator the instant release of the message is obvious. If a node has overheard exactly one retransmission it calculates its DFD according to the mathematically exact additional area it covers. In this special case (intersection of two circles) the computation is simple.

IV. Intensity-based Localization Algorithm

Existing localization algorithms ([1], [8]) depend on the possibility to distinguish two kinds of signals transmitted by an event. Thereby, the distance of the event is derived from the time difference of arrival (TDOA) of two different signals. For example, [1] uses the time difference of arrival between the shock wave and the muzzle blast generated by a gun.

From the time difference of arrival of these two signals the distance between the sniper and the measuring sensor node is calculated. In many cases the dependence on two different kinds of signals is restrictive and not easy to fulfill. In contrast to these algorithms, the algorithm discussed in the next section depends only on the intensity derived by a sensor node. This algorithm is generically computable and does not depend on predefined hardware what supplies a good degree of freedom.

A condition to determine the position (e_x, e_y) of an event E is that on any sensor node in the significance area the intensity determining the amplitude of the occurred event can be derived. We assume that the intensity ω_X derived at a sensor node X is related to the distance d_X the sensor node is away from an event, e.g the farther away a sensor node is, the lower its derived intensity is. This relationship is formalized in the following relation:

$$\omega_X \sim \frac{1}{d_X^\alpha} \qquad , \alpha > 1 \tag{1}$$

The exponent α in (1) affects the degree of attenuation of the measured intensity in dependence of the distance to the source of the event. (1) is a generalized formula of the acoustic, radio, etc. path loss models. The attenuation of an acoustic signal is for example similar to $\frac{1}{d^2}$.

It is crucial that the intensity cannot be used as a direct substitute of the distance in order to estimate the position of an event, but the square root of the ratio of the intensities of two sensor nodes is equal to the ratio of the distances of the two sensor nodes to an event. This will be shown in this section. Furthermore, we will show that if a sensor node A knows its own intensity ω_A and position (a_x, a_y) as well as the positions and intensities of at least three not collinear neighbor nodes B, C, D it can calculate the position of the event. The distance d_A of a sensor node A from the location of an event e can be calculated with the theorem of Pythagoras:

$$d_A^2 = (a_x - e_x)^2 + (a_y - e_y)^2 \tag{2}$$

From (2) and (1) we can derive the general equation to get the ratio of the intensities of two sensor nodes A and B:

$$\frac{(a_x - e_x)^2 + (a_y - e_y)^2}{(b_x - e_x)^2 + (b_y - e_y)^2} = \left(\frac{\omega_B}{\omega_A}\right)^{\frac{2}{\alpha}} \tag{3}$$

(3) means that the ratio of the distances from two sensor nodes A and B to the event location is equal to the ratio of the intensities derived on both nodes. It forms a circle, unless the ratio is 1. This case will be discussed later. As the position of the event E is contained on all three circles and the intersection point of the three circles is uniquely determined, the location of the event E is equivalent to the intersection point. This is true at least as long as the intensities derived at the sensor nodes are correct (an example generated with Maple is depicted in Fig. 6).

In order to prove the applicability of (3), we have to show that the denominator cannot be zero. This is however trivial as from (3) we can conclude that the denominator can only become zero if $b_x = e_x$ and $b_y = e_y$. This means the

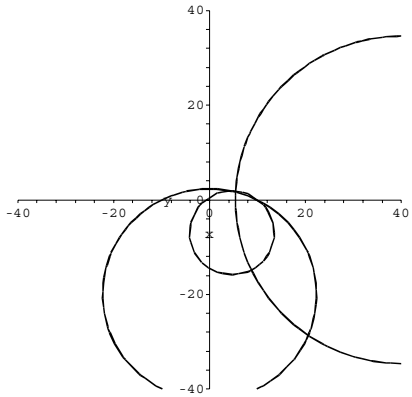


Fig. 6: Location of an event by intersection of three circles.

denominator can only be zero if the position of event E is exactly at the position of sensor node B. This case can be excluded, as the calculation of the position of an event is trivial if it occurs exactly at the location of a sensor node. Consequently, the position of an event is only calculated if it occurred not exactly at the location of one of the participating sensor nodes. In all these cases, the denominator cannot be zero.

Next, we calculate the intersection points of the circles that are derived from the ratios of the intensities of n non-collinear sensor nodes $S_1, .., S_n$, with $n > 1$. In order to facilitate the calculation, we set the point of origin of the coordinate system at the position of S_1 and the position of S_2 on the x-axis. This can be done without loss of generality. We will show that the calculation of the intersection point of the circles is equal to multilateration ([10], [12]). For better readability we replace ω_X^2 with ϕ_X. Using the ratios, we get the following equations:

$$\begin{aligned}
\frac{e_x^2 + e_y^2}{(s_{2_x} - e_x)^2 + e_y^2} &= \frac{\phi_{S_2}}{\phi_{S_1}} \\
\frac{e_x^2 + e_y^2}{(s_{3_x} - e_x)^2 + (s_{3_y} - e_y)^2} &= \frac{\phi_{S_3}}{\phi_{S_1}} \\
&\vdots \\
\frac{e_x^2 + e_y^2}{(s_{n_x} - e_x)^2 + (s_{n_y} - e_y)^2} &= \frac{\phi_{S_n}}{\phi_{S_1}}
\end{aligned}$$

If we dissolve the equations to zero and leave out the denominator from which we know that it cannot be zero , we get the following equations:

$$\begin{aligned}
\phi_{S_1}(e_x^2 + e_y^2) - \phi_{S_2}((S_{2x} - e_x)^2 + e_y^2) &= 0 \\
\phi_{S_1}(e_x^2 + e_y^2) - \phi_{S_3}((S_{3x} - e_x)^2 + (S_{3y} - e_y)^2) &= 0 \\
&\vdots \\
\phi_{S_1}(e_x^2 + e_y^2) - \phi_{S_n}((S_{nx} - e_x)^2 + (S_{ny} - e_y)^2) &= 0
\end{aligned}$$

The equations can be transformed to the following equations:

$$(\phi_{S_1} - \phi_{S_2})e_x^2 + (\phi_{S_1} - \phi_{S_2})e_y^2 + 2\phi_{S_2}S_{2x}e_x - \phi_{S_2}S_{2_x}^2 = 0$$
$$(\phi_{S_1} - \phi_{S_3})e_x^2 + (\phi_{S_1} - \phi_{S_3})e_y^2 +$$
$$2\phi_{S_3}(S_{3x}e_x + S_{3y}e_y) - \phi_{S_3}(S_{3_x}^2 + S_{3_y}^2) = 0$$
$$\vdots$$
$$(\phi_{S_1} - \phi_{S_n})e_x^2 + (\phi_{S_1} - \phi_{S_n})e_y^2 +$$
$$2\phi_{S_n}(S_{nx}e_x + S_{ny}e_y) - \phi_{S_n}(S_{n_x}^2 + S_{n_y}^2) = 0$$

The system can be linearized by subtracting the first equation from the last $n-1$ equations. Therefore, the first equation has individually to be multiplied with $\frac{\phi_{S_1}-\phi_{S_3}}{\phi_{S_1}-\phi_{S_2}}$,..., $\frac{\phi_{S_1}-\phi_{S_n}}{\phi_{S_1}-\phi_{S_2}}$. The resulting equations are subtracted from equations $2, ..., n$. In all $n-1$ resulting equations the unknown variables e_x, e_y are on the left side of the equations:

$$2\phi_{S_3}(S_{3x}e_x + S_{3y}e_y) - \frac{2\phi_{S_2}S_{2x}e_x(\phi_{S_1} - \phi_{S_3})}{\phi_{S_1} - \phi_{S_2}}$$
$$= \phi_{S_3}(S_{3_x}^2 + S_{3_y}^2) - \frac{\phi_{S_2}S_{2_x}^2(\phi_{S_1} - \phi_{S_3})}{\phi_A - \phi_{S_2}}$$
$$\vdots$$
$$2\phi_{S_n}(S_{nx}e_x + S_{ny}e_y) - \frac{2\phi_{S_2}S_{2x}e_x(\phi_{S_1} - \phi_{S_n})}{\phi_{S_1} - \phi_{S_2}}$$
$$= \phi_{S_n}(S_{n_x}^2 + S_{n_y}^2) - \frac{\phi_{S_2}S_{2_x}^2(\phi_{S_1} - \phi_{S_n})}{\phi_{S_1} - \phi_{S_2}}$$

The equations above indicate that $\phi_{S_1} \neq \phi_{S_2}$. The case of equality of ϕ_{S_1} and ϕ_{S_2} is discussed in the next paragraph. For now, we assume that $\phi_{S_1} \neq \phi_{S_2}$ and therefore neglect the denominator as soon as all terms are of the same denominator. If all terms are reordered, we get a system of linear equations of the form $Ax = b$, where

$$A = \begin{bmatrix} 2(\phi_{S_3}S_{3x}\Gamma + \phi_{S_2}S_{2x}(\phi_{S_3} - \phi_{S_1})) & 2\phi_{S_3}S_{3y}\Gamma \\ \vdots & \\ 2(\phi_{S_n}S_{nx}\Gamma + \phi_{S_2}S_{2x}(\phi_{S_n} - \phi_{S_1})) & 2\phi_{S_n}S_{ny}\Gamma \end{bmatrix}$$

$$b = \begin{bmatrix} \phi_{S_3}(S_{3_x}^2 + S_{3_y}^2)\Gamma - \phi_{S_2}S_{2_x}^2(\phi_{S_1} - \phi_{S_3}) \\ \vdots \\ \phi_{S_3}(S_{3_x}^2 + S_{3_y}^2)\Gamma - \phi_{S_2}S_{2_x}^2(\phi_{S_1} - \phi_{S_3}) \end{bmatrix}$$

For better readability we substituted $(\phi_{S_1} - \phi_{S_2})$ with Γ in A, b respectively. This system can be solved using a standard least-square approach: $E = (A^T A)^{-1}A^T B$, where E is the location estimation of the event. When the inverse matrix cannot be calculated, the location cannot be computed and the multilateration fails. This can happen if $\phi_{S_1} = \phi_{S_2}$. This is however no restriction, as in the case of $\phi_{S_1} = \phi_{S_2}$ the ratio of the intensities is 1 and the position of E lies on the vertical line through the middle of $\overline{S_1, S_2}$. The intersection of this vertical line with any of the participating circles results in the possible locations of event E. Consequently, in the case of $\phi_{S_1} = \phi_{S_2}$ the matrix is not calculated and the location is estimated using the intersection of the vertical line with any two independent circles derived from the intensities.

V. Simulations

In this section we present first simulation results of the broadcast as well as of the localization algorithm. To evaluate the two algorithms we implemented both in Matlab. All simulations were run over 20 seeds with a 95% confidence interval. The simulations described in this section share a common scenario. This standard scenario consists of 300 sensor nodes randomly distributed in a square area with sides of 100 meters. The radio range R of each sensor node is ten meters, which results in an average connectivity of nine neighbor nodes.

A. Evaluation of the DDB protocol

To evaluate the DDB broadcast protocol, we implemented the triangle metric along with the additional area metric and a simple flooding algorithm in Matlab. A detailed comparison of the DDB protocol to other broadcast protocols can be found in [13]. In this paper we will only investigate the performance of our new metric compared to the additional area coverage metric and a simple flooding protocol. It is to remark that the results we gained match well the results presented in [13] simulated with the Qualnet network simulator [15]. The thresholds for the protocols were chosen as follows: The additional area coverage (DDB_{AC}) metric uses a retransmission threshold RT of 22% of AC_{MAX}. This means a node using the DDB_{AC} metric cancels its retransmission when the additional area it covers with a rebroadcast is below 22% of AC_{MAX}. A value of 22% is chosen as the maximal loss of the DDB triangle ($DDB_{Triangle}$) approach is intrinsic 22%.

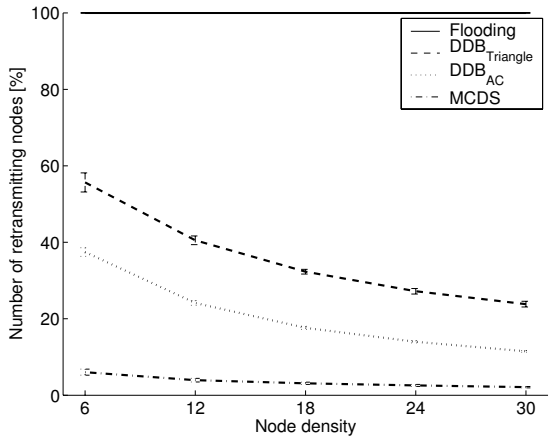


Fig. 7: Percentage of retransmitting nodes with different DDB metrics

To evaluate the performance of the different broadcast metrics, the network density is varied. The number of nodes in the network are altered from 300 up to 1100 in steps of 200 nodes resulting in an average number of neighbors from 9 to 33. The broadcast was always initiated by sensor node one. To have a benchmark for our algorithms we additionally implemented a minimum connected dominating set (MCDS) algorithm which computes the minimal set of nodes that is

necessary to reach all nodes within the network. The MCDS algorithm calculates the optimal solution for the broadcast problem, but is only computable with the knowledge about the whole network topology. The results of the DDB simulations are depicted in Fig. 7. The results of the delivery ratios of the different protocols are not depicted, as the delivery ratios are 100% in all simulations over all seeds. The DDB_{AC} as well as the $DDB_{Triangle}$ protocol decreased the number of retransmitting nodes in the network considerably, especially when the network density is high. Surprisingly DDB_{AC} needed almost 20% less retransmitting nodes than $DDB_{Triangle}$ over all node densities, resulting in less than 25% of retransmitting nodes as soon as the number of neighbors approximates 15 neighbors. The performance of the MCDS algorithm is by far the best. The difference can be explained by the retransmitting overhead of DDB for nodes close to the network area border. As the DDB algorithm is receiver based it has no knowledge about a possible area border and all nodes close to the border retransmit their message even as there are no additional nodes reachable.

We conclude that the DDB_{AC} protocol is the most suited protocol as long as the resource constraints on the nodes are not too restrictive. In other cases the $DDB_{Triangle}$ approach seems to be a reasonable alternative. A main advantage of the DDB protocol architecture is the absence of any states. The rebroadcast decision solely depends on the position of a node and of a function to assess that position. The results gained in these initial simulations encourage us to use the DDB protocol for the notification message distribution.

B. Evaluation of the ILA algorithm

In order to simulate noisy measurements on the sensor boards we implemented the inverse square law with a signal attenuation of $\frac{1}{d^2}$, where d is the distance, as error model. This model is for example appropriate for sound propagation. Intensity errors that are caused by noisy sensor readings are modeled according to the following formula:

$$err(\omega) = 1 \pm \lambda N(0,1) \cdot \omega \tag{4}$$

The error err depends on the derived intensity ω as well as the square distance d^2 between the event source and the measuring sensor node. Its amplitude is adjusted via the parameter λ. $N(0,1)$ is a normal distribution with mean zero and standard deviation one. According to (4) errors are normally distributed around the intensity whereas the amplitude of the deviation depends on the square distance a sensor node is away from an event and λ.

The simulations in this section share the common scenario parameters proposed above. Furthermore, a number of parameters are varied: number of sensor nodes, standard deviation of the measurement error, and reception radius. The event is always localized at position (50, 50). The reception radius D determines the distance until which the event is observable. In our simulations the reception radius varies from 10 m to 40 m. The amplitude of the standard deviation is adjusted over λ and its value varies from 0 to 25%. All simulations have in common that location estimations that are farther away from the calculating sensor node than the reception radius are discharged. This restriction is reasonable, as the event could not have been sensed by a sensor board if it is farther away than the reception radius. Very erroneous location estimations are seldom, but possible as a normal distribution is used in the error model, which permits very high deviations.

1) Influence of the distance on the accuracy: In these initial simulations we investigate the influence of the reception radius D. The location estimation is thereby performed on any sensor board in the reception area bounded by the reception radius and the location estimation error is averaged over all computations. We varied the reception radius accordingly to the values defined in the last section. In Fig. 8 the results of these simulations are depicted. The location estimation error is in all subsequent figures denoted in percentage of the radio range R.

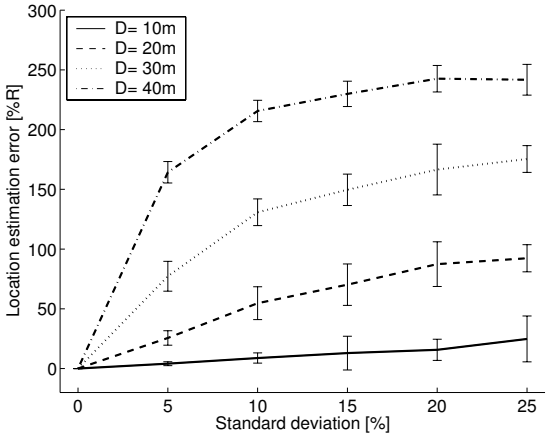


Fig. 8: Location error with variable reception radius D

The gained results indicate that the position estimation error is acceptable as long as the distance to the event is not too far and the sensor readings are not too noisy, i.e. λ is not too high. This result is not surprising and enforces our choice to elect the closest sensor node as winner and perform the location estimation on it. In the next simulation section we will explore the ILA performance if only the winner node performs the position estimation.

2) Position estimation at winner node: In this simulations we vary the number of nodes in the network to investigate the influence of the network density on the location estimation accuracy. 300, 500, and 1000 sensor nodes are simulated what results in an average connectivity of 9, 15, and 31 sensor nodes. We expect a better performance in denser networks, as the average distance between winner node and event source becomes smaller.

The results of the simulations are depicted in Fig. 9. The location estimation error is in all simulations between one meter and five meters. Thereby, the majority of the calculations supply location estimations less than two meters away from the exact event position. The large confidence intervals in these

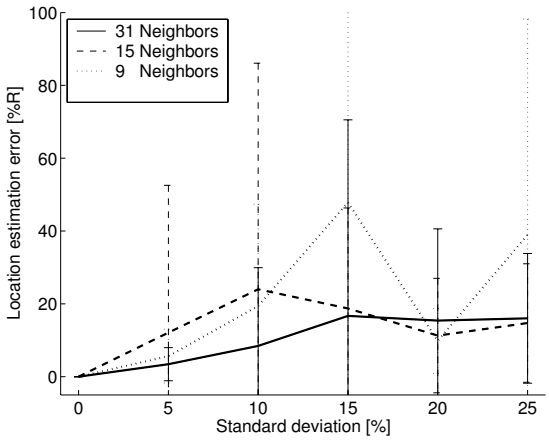


Fig. 9: Location estimation of winner

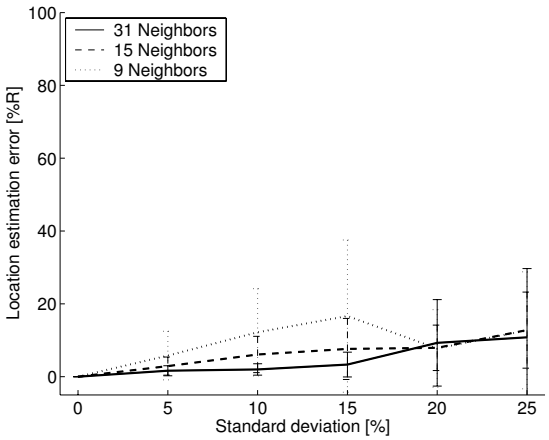


Fig. 10: Location estimation of winner including its vicinity

simulations indicate that the calculation limitation on only one sensor node is quite error-prone. This is substantiated by the sensibility of the ILA algorithm. The location estimation works well as long as there are no outliers among the measurements. On the other hand, if some measurements are very noisy an accurate event location estimation with only one sensor node fails, as the least square method used cannot handle very noisy sensor readings. We therefore will take the position estimations of the one-hop neighbor nodes of the winner also into consideration. The according refinements and simulations are discussed in the next subsection. Another possibility was to investigate other, less error-prone approaches to compute the location estimation. This remains to be done in future work.

3) Position estimation enhanced with information from winner vicinity: The results gained in the last section have shown that the location estimation on only one sensor board is in general not accurate enough. Consequently, we enhance the computation on the winner node with the position estimations calculated in its immediate vicinity. The necessary information is provided by the DEA algorithm proposed in Section III. The computation instruction is as follows: The winner node calculates the mean value and the standard deviation of its own position estimation as well as of the estimations of all of its neighbor nodes. It disregards then all estimations that are more than standard deviation away from the mean and computes the location with the remaining estimations. Furthermore, the location estimation fails if the standard deviation is more than half the mean value. This could happen in a noisy environment where a reliable location estimation is no longer given. Enhanced algorithms to operate in such scenarios will be investigated in future work.

The simulation results (see Fig. 10) show that the mean error as well as the standard deviation are considerably diminished with the algorithm proposed in this section. The node density influences the location estimation positively, but even with a rather low node connectivity of in average 9 neighbor nodes, we get feasible results. We conclude our evaluation with these first results, which indicate that a intensity-based localization on the winner node is possible. It moreover performed well in the simulations done so far. The results encourage us to continue our work and finally provide an efficient, distributed and accurate event detection and localization mechanism.

VI. Conclusions

We introduced two algorithms in this paper, namely the distributed election-winner notification algorithm (DENA) and the intensity-based localization algorithm (ILA). The combination of both algorithms builds a framework to efficiently and accurately detect and localize events. The dependence of the ILA algorithm on merely the derived intensity of an event on a sensor node constitutes its generic applicability as well as its weak binding on sensor hardware. In this paper we have shown that the performance of the ILA algorithm verified our expectations. It was shown that the location of an event can be computed in a distributed manner without need to gather any information on a sink node and that the accuracy of the event location improves the closer to the event the ILA algorithm is performed. Concerning the DENA algorithm, we have shown that the notification message is distributed efficiently by the dynamic broadcast protocol (DDB). The number of retransmitting nodes is decreased considerably with both DDB metrics tested so far. It remains to mention that both, the ILA and the DENA algorithm work close together in our approach. This is reasonable, it does however not restrict the applicability of the key functionality of both algorithms on their own.

VII. Future work

In future work we will further evaluate both algorithms. Special interest will be focused on the additional delay the DENA causes and on its ability to minimize the number of retransmitting nodes. At time, the DENA algorithm causes every node in the two-hop neighborhood of the winner node to respond. The possibility to query only a subset of these nodes will be considered. Furthermore, an implementation of the framework using the OMNeT++ [16] network simulator has been started. We will compare our framework to other

event detection and localization schemes. Thereby, we will focus on energy and bandwidth consumption. At time, the ILA algorithm needs the information of all neighboring nodes. In future work we will investigate if a subset of these neighbor nodes results in sufficiently accurate results. In that context, we will also investigate appropriate techniques to filter outliers. To deal with erroneous sensor measurements we currently use a linear mean-square approach. We will consider the usage of more sophisticated non-linear least square methods [17]. Finally, we will perform extensive sensor measurements on real hardware to obtain sophisticated error models and we will check if the algorithm is also feasible when applied on moving events and with multiple event sources.

REFERENCES

[1] G. Simon, G. Balogh, G. Bap, M. Maróti, B. Kusy, J. Sallai, A. Lédeczi, A. Nádas, and K. Frampton, "Sensor network-based countersniper system," in *SenSys*, Baltimore, Maryland, USA, November 2004.

[2] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 1, pp. 61–91, 2004.

[3] S. Guha, R. N. Murty, and E. G. Sirer, "Sextant: A unified node and event localization framework using non-convex constraints," in *MobiHoc'05*, Urbana-Champaign, Illinois, USA, May 2005, pp. 205–216.

[4] M. Kumar, L. Schwiebert, and M. Brockmeyer, "Efficient data aggregation middleware for wireless sensor networks," in *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, Fort Lauderdale, Florida, USA, October 25-27 2004, pp. 1579–1581.

[5] S. Li, S. H. Son, and J. A. Stankovic, "Event detection services using data service middleware in distributed sensor networks," in *ISPN'03*, Palo Alto, USA, April 2003, pp. 502–517.

[6] T. Abdelzaher, B. Blum, D. Evans, J. George, S. George, L. Gu, T. He, C. Huang, P. Nagaraddi, S. Son, P. Sorokin, J. Stankovic, and A. Wood, "Envirotrack: Towards an environmental computing paradigm for distributed sensor networks," in *Proc. of 24th International Conference on Distributed Computing Systems (ICDCS)*, Tokyo, Japan, Mar. 2004, to appear.

[7] N. Niculescu and B.Nath, "Ad hoc positioning system using aoa," in *Proceedings of IEEE INFOCOM Conference on Computer Communications*, San Francisco, CA, USA, 2003.

[8] S. Capkun, M. Hamdi, and J. P. Hubaux, "Gps-free positioning in mobile ad hoc networks," in *Proceedings of HICSS*, January 2001, pp. 3481–3490.

[9] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wälchli, "BLR: Beacon-less routing algorithm for mobile ad-hoc networks," *Computer Communications Journal*, vol. 27, no. 11, pp. 1076–1086, July 2004.

[10] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitive comparison," *Computer Networks*, vol. 43, no. 4, pp. 499–518, 2003.

[11] D. Niculescu, "Positioning in ad hoc sensor networks," *IEEE Communications Magazine*, vol. 18, no. 4, pp. 24–29, July/August 2004.

[12] A. Savvides, H. Park, and M. B. Srivastava, "The n-hop multilateration primitive for node localization problems," *Mobile Networks and Applications*, vol. 8, pp. 443–451, 2003.

[13] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bernoulli, "Optimized stateless broadcasting in wireless multi-hop networks," in *IEEE Infocom 2006*, Barcelona, April 23-29 2006, to appear.

[14] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, 1999, pp. 151–162.

[15] Qualnet, "Scalable network technologies (snt)." [Online]. Available: http://www.qualnet.com

[16] A. Varga, "Omnet++ discrete event simulation system." [Online]. Available: http://www.omnetpp.org/

[17] W. Navidi, W. S. Murphy, and W. Hereman, "Statistical methods in surveying by trilateration," *Computational Statistics and Data Analysis,*, vol. 27, pp. 209–227, 1998.

Content-Initiated Organization of Mobile Image Repositories

Bo Yang and Ali R. Hurson
Dept. of Computer Science and Engineering, the Pennsylvania State University, University Park, USA
{byang, hurson}@cse.psu.edu

***Abstract*—Considerable research has been done on the content-based image delivery and access in distributed repositories. As noted in the literature, there is always a tradeoff between the image quality and the access speed. In addition, the overall performance is greatly determined by the distribution of the image data, specially, in a heterogeneous environment. In this paper, a semantic-based image access scheme for a distributed, mobile, heterogeneous database infrastructure, the Ubiquitous Content Summary Model, is presented that addresses both the data quality and performance issues. With the ability of summarizing the content information and guiding the data distribution, the proposed solution is distinguished by its mathematical representation and concise abstraction of the semantic contents of image data, which are further integrated to form a general overview of a image data source and its application of word relationships to construct a hierarchical meta-data based on the summary schemas allowing imprecise queries. Furthermore, it achieves the optimal performance in terms of searching cost. The fundamental structure of the proposed model is presented.**

***Index Terms*—Mobile Image Retrieval, Content Distribution, Data Integration, Ontology Model**

I. Introduction

SEARCHING and accessing image data from a collection of heterogeneous mobile data sources such as sensor or ad hoc networks is becoming important in many applications. Undoubtedly, image information is among the most powerful representations of the human thought ─ representation of entities as objects and representation of the complex objects in term of simpler objects [6]. However, image data is also one of the most non-manipulative structures in computers [2]. Indexing on images is rather difficult, which makes accessing or semantically organizing image data more difficult to realize. In a heterogeneous distributed environment, the autonomy and heterogeneity of local databases introduce additional complexity to efficient representation and manipulation of image data.

Manuscript received September 19, 2005. This work was supported in part by the Office of Naval Research under contract N00014-02-1-0282 and National Science Foundation under contract IIS-0324835.

Bo Yang is with the Pennsylvania State University, PA 16802-6106 USA (phone: 814-863-3646; e-mail: byang@cse.psu.edu).

Ali R. Hurson is with the Pennsylvania State University, PA 16802-6106 USA (phone: 814-865-9505; e-mail: hurson@cse.psu.edu).

Traditionally feature vectors as the representative of image data are employed to facilitate content-based query processing [3, 8, 17]. For an application-specific domain, the features from image data, empirically or heuristically, are extracted, integrated, and represented as some vectors according to the predefined application criteria. Due to the application-specific requirements, this approach lacks scalability, accuracy, robustness, and efficiency.

As a better alternative, one can manipulate the image data at a unified semantic-entity level. In this approach, the heterogeneous image data sources are integrated into a unified format — in spite of their differences. In addition, in this unified paradigm, different types of media (audio, video, image, and text) can be considered as inter-convertible objects. With the thorough understanding of the image content, this paradigm provides the QoS-guaranteed query processing with higher scalability and lower resource requirements.

In this work, we present an iterative method to represent a image object as a combinatorial expression of its simpler objects. In addition, a novel content-aware image indexing and accessing scheme for a heterogeneous distributed database environment is discussed. The proposed scheme, as a target platform, is used and enhanced based on the proposed content-aware accessing scheme ─ Ubiquitous Content Summary Model (UCSM). With the guidance of these summaries, the content-based image retrieval scheme offers superior performance than several well-known image indexing schemes, as demonstrated in our experiments.

This paper is organized into seven parts: Section 2 briefly overviews the related work and background materials. Section 3 addresses the concepts of the UCSM. Section 4 introduces the methodology framework. Section 5 analyzes the performance of the proposed model. Section 6 further discusses the description of image data contents within the framework of enhanced UCSM. Finally, section 7 draws the paper to a conclusion.

II. Background and Related Work

2.1 Image Retrieval

As witnessed by the literature [3-12], the research on the content-based image retrieval processing has focused on three

interrelated issues:

- Representation of the image entities,
- Indexing and organization of the image entities, and
- Query processing strategies of image databases.

It should be noted that, most of the solutions that have advanced in the literature, study the image entities within the scope of the object oriented paradigm and hence, quantify a image data entity based on the features of its elementary objects.

Image data representation

The feature-based representation models can be further classified into four classes: the cluster-based organization, representative region approach, annotation-based organization, and decision tree-based organization [7-10].

The clustering-based approach partitions the image data objects into clusters of semantically similar objects [7, 20- 24]. The clustering approach can be further grouped into the supervised and unsupervised mode [7, 23, 24]. The supervised clustering approach utilizes the user's knowledge as input to cluster image objects, and hence it is not a general purpose clustering approach. As expected, the unsupervised clustering approach does not need the interaction with user. Hence, it is an ideal mechanism to cluster unknown image data automatically. Alternatively, the representative region approach, according to the Expectation Maximization (EM), constructs a simple description of the image objects based on several of the most representative regions of the objects [8, 25]. Motivated by the text attachment of image objects, the annotation-based organization paradigm makes use of manually or automatically added annotations [9]. Finally, integrated with the relevance feedback, the decision-tree-based approach organizes image data in a hierarchical fashion that separates the data by recursively applying decision rules [10, 26, 27, 28].

Efficient indexing scheme

Employment of efficient indexing is the key issue to the real-time retrieval of the image data . Efficient indexing is relatively more complicated when heterogeneous image data sources need to be integrated together. Two classes of indexing schemes have been discussed in the literature: The partition based indexing and the region-based indexing.

The partition-based indexing scheme, Quad-tree [11, 29], K-d-tree [15, 30], and VP-tree [16, 31], is a top-down process that recursively, divides the image object (or multidimensional feature space) into disjoint partitions while constructing a hierarchical data structure that represents the index of the image object. The region-based indexing scheme, R-tree [12], R*-tree [13], and SR-tree [14], takes a bottom-up approach in forming an access index ─ regard the image data as point objects in multi-dimensional feature space, employ some small regions to cover all the points, and then recursively, combines small regions into larger groups (how does the index form).

Efficient query processing

Searching for the image data is the crucial step in providing real-time image services. Based on the type of information submitted to the search engine, three searching strategies have been recognized: keyword querying, example matching, and fast browsing [18]. Cox et al. [18] proposed a Bayesian image retrieval system that accommodated all these three strategies.

Within the scope of a networked environment, the literature has addressed several practical image systems. The IBM Query by Image Content project (QBIC System by IBM Almaden Research Center) [3] allows users to query an image collection using features of image content – colors, textures, shapes, locations, and layout of images and image objects. Multi-dimensional feature vectors are employed to describe image content, with an R*-tree as the indexing structure. Speech Recognition (Jabber experimental system) [4] uses concept clustering based on indexing on audio content of a videoconference. It employs word recognition facility to set up an index based on the recognized words. To find the main topics and make a meaningful index, the Jabber system uses several lexical conglomerates, such as chains, trees, and clusters. The system uses surrounding words as restrictions, then compares the semantic distances of different relationships, and finally determines the relationship with minimal distance as the meaning of specified word. The Photobook System (developed by the MIT Media Lab) [5] is a system for Face recognition based on eigenvector descriptor. The Photobook System efficiently uses "distance-from-feature-space" (DFFS) to detect eigen-features. Given an input image, a feature distance-map is built by computing the DFFS at each pixel. The global minimum of this distance map is then selected as the best feature match.

In spite of the progress reported in the literature, the content-based image research is in its infancy . The scope of research in image database extends drastically when parameters such as autonomy, heterogeneity, mobility, and wireless limitation are added to the mix [6].

2.2 Multi-Database Systems

A multi-database system ***MDBS*** is a distributed system that acts as a global layer sitting on top of multiple preexisting distributed, autonomous, and heterogeneous local databases $\{\boldsymbol{LDBS}_i, \text{ for } 1 \le i \le n\}$ [1]. The local databases are connected via wired/wireless networks to form a global information sharing system. The local databases play dual roles in managing the data sources: On one hand, each local database $\boldsymbol{LDBS}_i$ located at site $\boldsymbol{LS}_i$, manages its local dataset $\boldsymbol{LDS}_i$. On the other hand, all local databases are harmonized under the restriction of a global access control mechanism. Figure 1 depicts a multi-database system.

Two types of requests exist in a multi-database system: local requests and global requests. The local requests are performed by the local database systems autonomously. The global requests, however, require the cooperation among local data-

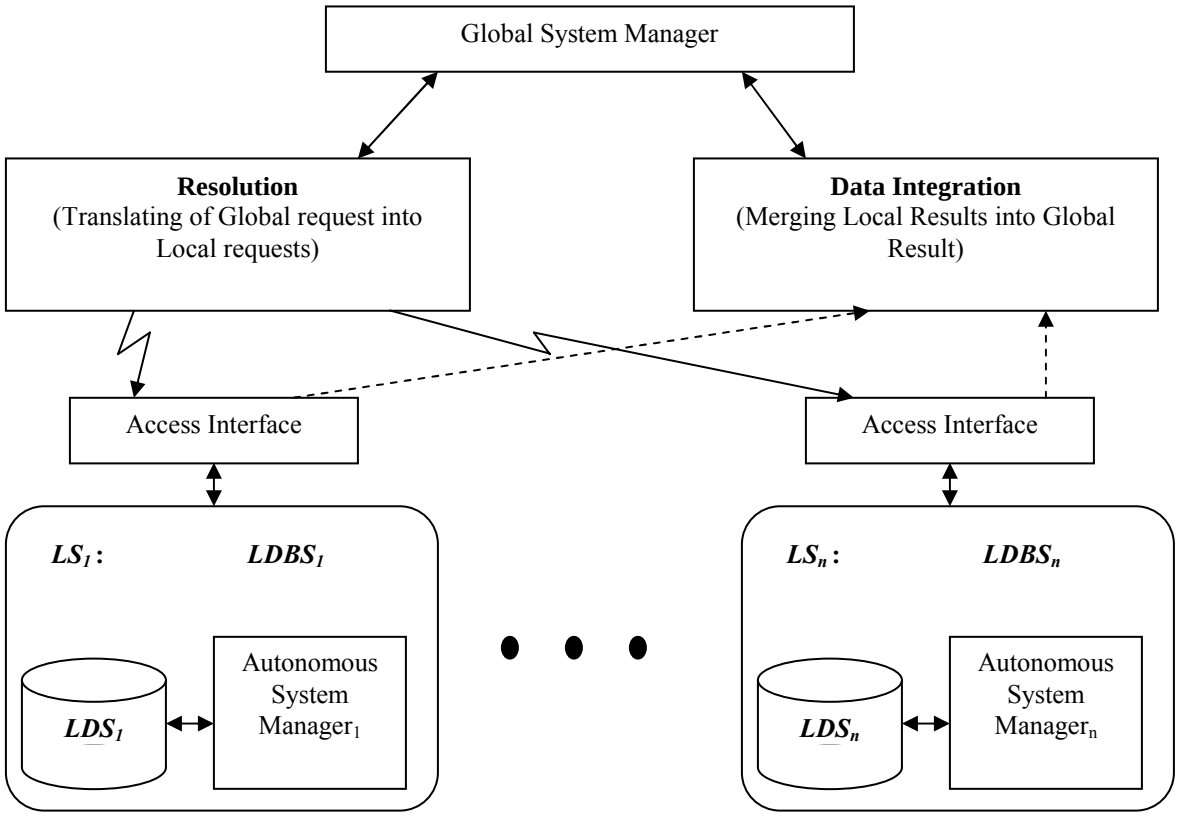


Figure 1: The Multi-database System.

bases. Normally, a global request R^g is the combination of a set of sub-requests $\{R^g_{[i]}$, for $1 \leq i \leq m\}$, where each sub-request $R^g_{[i]}$ is treated as a local request that can be executed on one of the local databases. The global request R^g is complete only after all the sub-requests are terminated at the local databases.

2.3 Summary Schemas Model

The Summary-Schemas Model (SSM) was proposed as a solution to large-scale multi-database systems [1, 19, 32]. It is a content-aware infrastructure that enables imprecise query processing on distributed heterogeneous data sources. The scalable content-aware query processing is made possible with the aid of an indexing meta-data based on the hierarchy of summary schemas, which comprises three major components: a thesaurus, a collection of autonomous local nodes, and a set of summary-schemas nodes.

The thesaurus provides an automated taxonomy that categorizes the local access terms and defines their semantic relationships ─ the thesaurus may utilize any of the off-the-shelf thesauruses (e.g. Roger's Thesaurus) as its basis. A Semantic-Distance Metric (SDM) is defined to provide quantitative measurement of "semantic similarity" between terms [19]. A local node is a physical database containing the data sources in different forms and representation, i.e., image data, textual data, formatted data. The local node is organized autonomously, on condition that its semantic content is communicated to the global mechanism at the thesaurus level. With the help of the thesaurus, the local access terms are classified, mapped, and integrated to their hypernyms. A summary schemas node is a virtual database concisely describing the semantic contents of its child (children) node(s). More detailed descriptions about the SSM can be found in [1, 19, 32].

In contrast with other multi-database solutions, the SSM has the following properties due to its unique semantic-based organization:

- The SSM allows automatic semantic-based data integration regardless of the heterogeneity of data sources.
- The SSM allows imprecise query processing. As a result, a user is able to submit his/her request in a free format notation.
- The SSM provides highly efficient content-based indexing capability.
- The SSM offers high scalability and robustness.

3 PRELIMINARIES

To overcome the aforementioned shortcomings of existing image systems, we introduced a novel image access paradigm based on the summary schemas model (SSM). As a scalable content-based scheme, the SSM prototype was originally proposed to resolve the name differences among semantically similar data in multi-database systems. Due to its concise

structure and strong cross-modal representation capability, the SSM provides an efficient method of accessing image data.

3.1 Representation of Image Objects

The foundation of most image retrieval systems is the feature representation of image objects [2]. Extracted in the preprocessing stage of image representation, the features play an important role in quantizing the non-structured image data. The following definitions are the fundamental concepts of image retrieval systems.

Definition 1: Feature extraction

Assume $\boldsymbol{I} = \{I_j \mid 1 \le j \le n\}$ is a set of image objects, and $\boldsymbol{\Phi} = \{\varphi_i \mid 1 \le i \le m\}$ is the ordered mask of feature extraction priorities. The feature extraction process is a function $\boldsymbol{f}$: $\boldsymbol{I} \times \boldsymbol{\Phi} \rightarrow \boldsymbol{D}$, where $\boldsymbol{D}$ is the feature destination set. $\boldsymbol{D}$ could be a set of high-dimensional vectors, a set of cluster IDs, or real numbers indicating the semantic cluster that the image object belongs to.

In the image retrieval systems, there are two types of features: granule-level features and object-level features. The granule-level features are derived from the original format of image storage — i.e., those characteristics that directly or indirectly are obtained from the pixels, such as colors, textures, saturation. The object-level features, in contrast, are obtained from the recognition of the higher-level understanding of the image data — the semantic topics of the image data. In the aforementioned image retrieval system, the object-level features can be the recognized as elementary data items, shapes, spatial relationship, and etc.

Definition 2: Semantic distance

Suppose $\boldsymbol{I} = \{I_j \mid 1 \le j \le n\}$ is the set of image objects, and $\boldsymbol{\Phi} = \{\varphi_i \mid 1 \le i \le m\}$ is the ordered mask of feature extraction priorities. The semantic distance on feature φ_i is a function $\boldsymbol{g}^{\varphi_i}$: $\boldsymbol{I} \times \boldsymbol{I} \rightarrow \boldsymbol{R}$, where $\boldsymbol{R}$ is the set of real numbers. The semantic distance function $\boldsymbol{g}^{\varphi_i}$ compares two image objects and returns their semantic distance.

The function $\boldsymbol{g}^{\varphi_i}$ satisfies the following characteristics:

1) *For any pair of image objects x and y:* $\boldsymbol{g}^{\varphi i}(x, y) \ge 0$,
2) $\mathbf{g}^{\varphi i}(x, y) = 0$ iff $x = y$,
3) *For any pair of image objects x and y:* $\mathbf{g}^{\varphi i}(x, y) = \mathbf{g}^{\varphi i}(y, x)$, *and*
4) *For image objects x, y, and z:* $\mathbf{g}^{\varphi i}(x, y) + \mathbf{g}^{\varphi i}(y, z) \le \mathbf{g}^{\varphi i}(x, z)$.

The semantic distance provides a quantized measure of comparing the difference between image objects. Based on the definition of semantic distance, we introduce the nearest neighbor concept that is widely used in most image retrieval systems.

Definition 3: The 1-nearest neighbor

Assume $\boldsymbol{I} = \{I_j \mid 1 \le j \le n\}$ is the set of image objects, $\boldsymbol{\Phi} = \{\varphi_i \mid 1 \le i \le m\}$ is the ordered mask of feature extraction priorities, $\boldsymbol{W} = \{\boldsymbol{w}_i \mid 1 \le i \le m\}$ is the set of weights of the feature extraction priorities, and $\boldsymbol{X}$ is the image object that is used as the query example. The nearest-neighbor searching process is a function $\boldsymbol{Q}$:

$$\boldsymbol{Q}(\boldsymbol{X}, \boldsymbol{I}, \boldsymbol{\Phi}, \boldsymbol{W}) = \{\boldsymbol{I_i} \mid \boldsymbol{I_i} = min\{\textstyle\sum_{k=1}^{m}(\boldsymbol{g}^{\varphi k}(\boldsymbol{X}, \boldsymbol{I}_j) * \boldsymbol{w}_k)\}_{j=1}^{n}\}$$

Definition 4: The K-nearest neighbor

Assume $\boldsymbol{I} = \{I_j \mid 1 \le j \le n\}$ is the set of image objects, $\boldsymbol{\Phi} = \{\varphi_i \mid 1 \le i \le m\}$ is the ordered mask of feature extraction priorities, $\boldsymbol{W} = \{\boldsymbol{w}_i \mid 1 \le i \le m\}$ is the set of weight of the feature extraction priorities, $\boldsymbol{K}$ is the parameter indicating the number of nearest neighbors, and $\boldsymbol{X}$ is the image object that is used as the query example. The K-nearest-neighbor searching process is a function $\boldsymbol{Q}^*$:

$$\boldsymbol{Q}^*(\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{I}, \boldsymbol{\Phi}, \boldsymbol{W}) = \{\boldsymbol{I_i} \mid |\boldsymbol{Q}^*(\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{I}, \boldsymbol{\Phi}, \boldsymbol{W})| = \boldsymbol{K},\ \forall \boldsymbol{I}'_j \notin \boldsymbol{Q}^*(\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{I}, \boldsymbol{\Phi}, \boldsymbol{W}),\ \textstyle\sum_{k=1}^{m}(\boldsymbol{g}^{\varphi k}(\boldsymbol{X}, \boldsymbol{I}_i) * \boldsymbol{w}_k) \le \sum_{k=1}^{m}(\boldsymbol{g}^{\varphi k}(\boldsymbol{X}, \boldsymbol{I}'_j) * \boldsymbol{w}_k)\}$$

The 1-nearest-neighbor search returns the image objects with the smallest semantic distance from the query example. The K-nearest-neighbor search returns K image objects, with the decreasing order of their similarities to the query example.

Based on the definition of semantic distance, the nearest-neighbor search can be performed in a multi-dimensional space of features. If we consider each feature as a dimension, the image object can be considered as a vertex in the multidimensional space of features. In this multidimensional space, the semantic distance between image objects is quantified as the spatial distance between vertices. The nearest neighbors should have similar positions as the querying example object. In another word, the nearest neighbors resides within a sphere whose center is the querying image object (Figure 2). In Figure 2, the semantic distance between any nearest neighbor and the querying image object is less than the radius of the sphere.

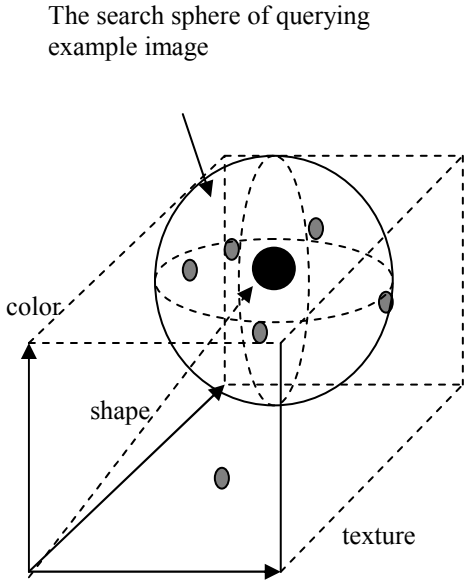


Figure 2: The search sphere for nearest neighbors.

Definition 5: The Elementary Entity

The elementary entities are those data entities that semantically represent basic objects (objects that cannot be divided further). Formally, the semantic contents of an elementary entity (E) can be considered as a first-order logic expression.

Let $E = f_1 \wedge f_2 \wedge \ldots \wedge f_n$, where $f_i = p_{i1} \vee p_{i2} \vee \ldots \vee p_{im}$ is the disjunction of some logic predicates (true/false values) and $p_{i1} \ldots p_{im}$ form a logic predicate set F_i, ─ In the feature-based image data sets, f_i indicates the i^{th} feature of the elementary entity. The semantic contents of an elementary entity can then be defined as:

$$E = \bigwedge_{i=1}^{n} (\bigvee_{j=1}^{m} p_{ij}), \qquad \text{for every } p_{ij} \in F_i.$$

Note that in any term $f_i = p_{i1} \vee p_{i2} \vee \ldots \vee p_{im}$, there is one and only one true predicate p_{ij}.. For instance, if $p_{i1}, p_{i2} \ldots p_{im}$ correspond to all possible color patterns, the semantic content of f_i at any time is a specific color pattern. Since f_i is disjunction of $p_{i1}, p_{i2} \ldots p_{im}$, the false predicates do not affect the final result. The content of an elementary entity is restricted by its conjunction terms $f_1, f_2 \ldots f_n$, which are the extracted features in application domains.

Definition 6: The Image Object

A image object is a collection of elementary entities. Given the above definition of elementary entities $E_1, E_2, \ldots, E_k$, the semantic contents of a image object can be defined as:

$$S = \bigcup_{i=1}^{k} E_j .$$

According to the definitions 5 and 6, a image object is considered as a combination of logic terms, whose value represents its semantic content. The analysis of semantic contents is then converted to the evaluation of logic terms and their combinations. This content representation approach offers the following advantages:

- The logic terms provide a convenient way to describe semantic contents concisely and precisely ─ The features of any elementary image object can be determined easily, hence, the semantic content of a complex image object can also be obtained using logic computations. The similarity between objects can be considered as the equivalence of their corresponding logic terms.
- This logic representation of image content is often more space efficient than feature vector. In a specific image database system, the feature vector is often of fixed length to facilitate the operations. However, for some objects, some features may be null. Although these null features do not contribute to the semantic contents of image objects, they still occupy space in the feature vectors, and hence lower space utilization. In contrast, the logic representation can improve storage utilization by eliminating the null features from logic terms.
- Compared with feature vectors, the logic terms provide an understanding of image contents that is closer to human perception.
- Optimization can be easily performed on logic terms using mathematical analysis. By replacing long terms with mathematically equivalent terms of shorter lengths, the image representation can be automatically optimized.
- Based on the equivalence of logic terms, the semantically similar objects can be easily found and grouped into the same clusters. This organization facilitates the nearest-neighbor retrieval, and at the same time reduces overlapping and redundancy. Hence, searching efficiency and storage utilization are both improved.

An interesting issue is that the logic representation approach can be seamlessly integrated with SSM. Each concept in the logic representation can find its counterpart in SSM, and hence can be performed within the domain of SSM. For instance, the equivalence between logic terms can be considered as the synonym relationship between summary schemas. Hence, the operation of finding equivalent terms in logic domain can be mapped to searching for synonyms in summary schemas domain. Similarly, other relationships between logic terms can also be conveniently represented in SSM. If term A is equal to a part of term B, then this "inclusion" relationship between A and B can be described with hypernym and hyponym relationships in SSM. Considering the strengths of SSM in organizing data [1, 19], we incorporate the logic representation within the framework of SSM. For simplicity, we consider logic term and summary schema as the same concept in the remaining part of this paper.

3.2 The Rationale of UCSM

The UCSM, the integration of SSM with logic predicate representation of image data contents, takes the aforementioned concepts further to a more general representation of image contents, with a series of interrelated access terms representing the relationship among image objects at the data base granularity level. As noted before, the SSM organizes and classify the data sources based on database semantics – database schemas and summary schemas, and word relationships. Within the SSM terminology then, a database schema is a group of access terms that describe the contents of image data source. A summary schema is a concise abstract of semantic contents of a group of database schemas. The summary schemas are connected through synonym, hypernym, and hyponym links. These links are logically used to represent the semantic relationship among image data objects.

Synonym links in the UCSM hierarchy are used to represent the semantically similar data entities regardless of their representation and/or access term differences. Refer to our methodology; a synonym link represents equivalence relationship between two image (or two groups of image) data objects. For instance, assume in an image environment, photos are grouped according to their authors. As a result, two similar photos taken by different authors will be kept in different databases. To represent similarity relationship between these two photos kept in two different databases, the UCSM employs synonym links to connect and group the similar image objects together.

- A hypernym is the generalized description of the common characteristics of a group of data entities. For instance, the hypernym of dogs, monkeys, and horses is mammal. To find the proper hypernyms of image

objects, the UCSM maintains an on-line system taxonomy that provides the mapping from image objects to hypernym terms. Based on the hypernyms of image objects, the UCSM can generate the higher-level hypernyms that describe the more comprehensive concepts. For example, the hypernym of mammals, birds, fish and reptiles is animal. Recursive application of hypernym relation generates the hierarchical meta-data of the UCSM. This in tern conceptually gives a concise semantic view of all the globally shared image objects. Refer to our methodology; a hypernym link represents "a member of" relationship.

- A hyponym is the counter concept of a hypernym. It is the specialized description of the precise characteristics of image objects. It inherits the abstract description from its direct hypernym, and possesses its own particular features. The UCSM uses hyponyms links to indicate the hyponyms of every hypernym. These links compose the routes from the most abstract descriptions to the specific image objects.

One of the merits of the UCSM is the ease of nearest-neighbor search operation. In the UCSM, the nearest neighbors are considered as synonyms that are connected through synonym links. As a result, the nearest-neighbor search is simplified into a process of finding the synonym links. In other indexing models, the nearest neighbor indexing is a time-consuming process that requires searching through a subset of the distributed image databases [13, 14, 16].

3.3 The Structure of Summaries

The heart of the UCSM prototype is the generation of summary schemas, which imply the semantic content of image objects. Motivated by the observation that the low accuracy of the present image retrieval systems is due to the improper selection of granule-level features as the representation foundation, UCSM prototype employs the object-level features obtained from some computer vision algorithms [17, 25]. Since these object-level features usually have stronger descriptive capability than granule-level features, the summary schemas are able to describe the semantic contents of image data using more concise terms.

To represent the content of image objects in a computer-friendly structural fashion, the UCSM organizes the image objects into layers according to their semantic contents. A image object, say, an image, can be considered as the combination of a series of elementary entities, such as animals, vehicles, and buildings.

4 PROPOSED METHODOLOGY

Based on the summary schemas topology, the image databases are organized in a hierarchy, which consists of leaf nodes and intermediate summary schema nodes. The leaf nodes, containing the real data, are clustered according to their semantic contents. The common information of each group is extracted and kept in a higher-level summary schema node. This semantic summarization process continues until it reaches the root node. Consequently, the root node keeps the most abstract view of all the globally shared data objects. Traversal from root node to leaf nodes, the UCSM hierarchy provides a gradually refining method to find the image objects.

This UCSM hierarchy is notable in its strong support to content-based image retrieval. The query can be submitted at any summary schema node as well as the local databases. The query is resolved as a series of matching query requirements with summary schemas. The query processor first compares the summary schema's entry at the query origin node with the query goal. In case of a successful match, then the query processor returns the accessing terms as the result, if the query is originated at a leaf node, or query goal is sent to the proper child (children) node (s). Otherwise, the query goal moves up the summary schema's hierarchy and tries to match the query at a higher level. This process continues until the query goal reaches leaf node (s) or the query goal reaches root without a successful match. Based on our experimental result [32], the height of the UCSM hierarchy is short, which by default implies efficient search process.

In the UCSM, a user could issue either imprecise or precise queries. Imprecise queries are those that may have access terms different from local access terms and/or may not specify any location of the data. Precise queries, on the other hand, use exact local access terms and also give specific data location. A precise query can be resolved by sending the query directly to the specified database whereas the process of resolving an imprecise query is more involved in identifying semantic intents of the user's query and then, based on that intention, the query shall be resolved.

The hierarchical structure of the UCSM is used to resolve imprecise queries. The query resolution starts at the node issuing the query. Each term 'a' in the query is compared with the terms in the schema 's' at that node. If the SDM between all query terms and schema terms is less than or equal to some specified threshold SDM, the query is resolved either at that node (if it is a local node) or at the children of that node (if it is an SSM node). On the other hand, if 'a' and 's' are not linguistically related; hence, not matched, the search proceeds to the parent of the current node. This process will recursively continue until either the search reaches the top of the UCSM hierarchy and fails with no possible downward search, or the search fails at a particular node on a downward traversal, or the search reaches a local-node where the query is resolved. The search fails at a specific node when the query terms do not match the schema terms at that node.

Given a random set of image objects in a heterogeneous multi-database environment, the UCSM prototype relies on its summarization capability to construct a hierarchical indexing structure for these objects. Hence, finding proper content integration methods is the crucial step to show the effectiveness of the UCSM. Two classes of content integration are employed in the UCSM framework:

- Replacing a set of specific terms with a more general term (hypernym relation), such as summarizing "car",

"bus", and "truck" into a more abstract concept "auto"; and

- Reorganizing combinations of features to a more concise description, such as changing {[(object = dog) ∧ (color = grey)] ⋃ [(object = dog) ∧ (color = white)]} into a shorter equivalent term {(object = dog) ∧ [(color = grey) ∨ (color = white)]}.

The first type of content integration is automated and relies on a system thesaurus [19, 32]. The second type, however, is an intriguing new issue that has not been explored. This content integration process, if resolved with properly designed strategies, would drastically reduce the cost of content-based retrieval in image databases.

Our goal in the content integration process is to specify the hidden semantic relationships among the image objects using an effective analytical comparison of the features. Inspired by the formation of Karnaugh Maps, we designed a combinatorial optimization table to shorten the complex combinations of features into condensed logic terms.

A UCSM-based indexing hierarchy is constructed during this content integration process. Compared with other indexing models, the UCSM hierarchy provides a more efficient content description by exploiting the unique summary representation of image objects. Our experimental results show that the UCSM has superior performance than some classic image indexing models, such as R*-tree and M-tree.

5 Theoretical Study

In this section, the performance of the proposed UCSM-based searching scheme is analyzed. As it is expected an effective content-based retrieval mechanism requires the ability to capture the semantic contents of the data objects accurately and efficient data searching. We analyze the performance of the UCSM based on two performance metrics; the size of the summary schemas and the searching cost in the summary-schemas hierarchy. Some presumptions are given to simplify the analysis process and final conclusions. The rationality of performance analysis is further supported by our simulation results.

5.1 The Analysis of Summary Schemas

In section 4, the semantic contents of image objects were mapped to a multidimensional space of features, then expressed as the disjunction of some first-order logic terms, and finally converted to a concise representation with the help of a combinatory optimization table. We now justify the rationality of summary schemas by showing that the size of the summary schemas is drastically shortened after optimization.

The size of summary schemas is measured by the number of predicates, which is comparable with the number of features in most of the other content-based indexing models. Reducing the number of predicates can reduce the number of comparisons in image object matching and consequently the communication cost during the query processing.

We assume a image object (say, an image) I having K elementary entities $E_1, E_2, \ldots, E_k$. Each elementary entity is within the multidimensional feature space indicated by $f_1, f_2, \ldots, f_n$, where $f_i = p_{i1} \vee p_{i2} \vee \ldots \vee p_{im}$ is the disjunction of some logic predicates. As mentioned in section 2, the semantic content of the image object I can be represented as the union of the elementary entities, which are expressed as the conjunctions of predicates. Refer to *Definitions 5* and *6*, we have the following expression of semantic content:

$$S = \bigcup_{i=1}^{k} E_i = \bigcup_{i=1}^{k} [\bigwedge_{j=1}^{n} (\bigvee_{h=1}^{m} p_{jh})], \text{ for every } p_{jh} \in F_j$$

Since the semantic content of feature f_i is uniquely determined by the true predicate p_{ix} within $p_{i1}, p_{i2}, \ldots, p_{im}$, we change the above equation into a simpler form:

$$S = \bigcup_{i=1}^{k} (\bigwedge_{j=1}^{n} p_j^{(i)})$$

where $p_j^{(i)}$ is the true predicate of the *jth* feature of the *ith* elementary entity.

Let S^* be the final result from the combinatory optimization table. Given the definition of combinatory optimization table, S^* by default expresses the same semantic content as S. According to step 4 of the optimization method, S^* is the union of a collection of clusters $C_1, C_2, \ldots, C_q$, with each cluster indicating several elementary entities. Hence, S^* can be expressed as the following:

$$S^* = \bigcup_{i=1}^{q} C_i.$$

As mentioned earlier, each cluster corresponds to a rectangular region in the combinatory optimization table. Assume cluster C_i is horizontally indicated by labels $L_1', L_2', \ldots, L_r'$, and vertically indicated by labels $L_1'', L_2'', \ldots, L_s''$. Here any label in $L_1', L_2', \ldots, L_r'$ or $L_1'', L_2'', \ldots, L_s''$ can be the conjunction of several predicates in equation (2). For instance, L_1' may be (object = cat) ∧ (color = grey). Then C_i can be expressed as $(L_1' \vee L_2' \vee \ldots \vee L_r') \wedge (L_1'' \vee L_2'' \vee \ldots \vee L_s'')$, or $\bigvee_{i=1}^{r} [\bigvee_{j=1}^{s} (L_i' \wedge L_j'')]$.

When representing the clusters with labels, if a cluster is a whole row/column, then the label for the row/column can be omitted in the representation. For instance, if all texture patterns are in a cluster, then this cluster does not need the feature "texture" in its representation. For the clusters that do not contain whole rows/columns, avoiding overlapping with other clusters can reduce the size of summary schemas.

5.2 The Search Cost

Some content-based indexing models evaluated searching cost in terms of the number of comparisons [18], while others use the number of disk accesses as the searching cost [14, 16]. We believe that both parameters should be accounted when determining the searching cost. In this section, the searching cost of the summary-schemas hierarchy is calculated as the average number of accesses at the summary-schemas nodes

(number of comparisons) and local nodes (number of disk accesses).

We assume a set of n image objects, $I_1, I_2, \ldots, I_n$ and the following notations in our analysis:

- $P(I_i)$: The probability of being queried for image object I_i.
- $\overline{W}$: The average searching cost for all image objects in any indexing tree model.
- $W(I_i)$: The searching cost for image object I_i in any indexing tree model.
- $N(I_i)$: The number of nodes on the path from root node to image object I_i in any indexing tree model.
- $\overline{W}^*$:The average searching cost for all image objects in summary schemas model.
- $W^*(I_i)$: The searching cost for image object I_i in summary schemas model.
- $N^*(I_i)$: The number of nodes on the path from root node to image object I_i in summary schemas model.

Given the above notations, the searching cost for a request composed on n random objects is:

$$\overline{W} = \sum_{i=1}^{n} [P(I_i)\, W(I_i)] \qquad (1)$$

Considering the definitions of the indexing models [11-18], the content-based searching always starts from the root node, traverses within the indexing tree, and finally arrives at the image object I_i. Thus,

$$W(I_i) \geq N(I_i) \qquad (2)$$

$$\overline{W} = \sum_{i=1}^{n} [P(I_i)\, W(I_i)] \geq \sum_{i=1}^{n} [P(I_i)\, N(I_i)] \qquad (3)$$

Lemma 1: The UCSM hierarchy does not contain any form of overlapping between its branches.

The elimination of overlapping between branches of the UCSM hierarchy is due to the existence of synonym links. While the other indexing models (R-tree family, SS-tree, etc.) are striving for the reduction of overlapping, the UCSM hierarchy can completely remove the overlapping data by adding some synonym links to other branches.

Proposition 1: Given a fixed set of image objects, the UCSM hierarchy has less or equal height than any indexing tree.

Proof. We will prove that any indexing tree can be described using the UCSM hierarchy with less or equal height. Given any arbitrary set of image objects $\boldsymbol{I} = \{I_1, I_2, \ldots, I_n\}$ and any indexing tree model $\boldsymbol{M}$, we can construct an equivalent UCSM hierarchy in the following way:

Let $\boldsymbol{T}$ be the indexing tree generated from applying indexing model $\boldsymbol{M}$ to the image data set $\boldsymbol{I}$. And for any node $\boldsymbol{n_i}$ in tree $\boldsymbol{T}$, let ***feature*** $(\boldsymbol{n_i})$ be the set of features that globally identify node $\boldsymbol{n_i}$, ***parent*** $(\boldsymbol{n_i})$ denote the parent node of $\boldsymbol{n_i}$, and ***children*** $(\boldsymbol{n_i})$ be the set of child (children) node(s) of $\boldsymbol{n_i}$.

First, we group the leaf nodes into clusters $\boldsymbol{C_1, C_2, \ldots, C_k}$ according to common parents. For any cluster $\boldsymbol{C_j}$, make a union of all features of the nodes in this cluster to get the features for the common parent node. That is to say, suppose $\boldsymbol{n^*}$ is the common parent node of cluster $\boldsymbol{C_j}$, ***feature*** $(\boldsymbol{n^*}) = \bigcup_{ni \in Cj}$ ***feature*** $(\boldsymbol{n_i})$. The rationale behind this union is the fact that any node in the tree-based indexing structure can be identified by the route from the root to that node, which can also be determined by the features available at that node.

Next, we can use the aforementioned summarization process to generate a proper summary schema for the parent node $\boldsymbol{n^*}$. By recursively making abstraction, we construct a UCSM hierarchy with no more height than the indexing tree $\boldsymbol{T}$. According to *Lemma 1*, this UCSM hierarchy does not contain any overlapping, which may further reduce the height of the UCSM hierarchy. Hence, the UCSM hierarchy can describe any feature-based indexing tree with less or equal height. Or in another word, for any image object I_i, we have $N^*(I_i) \leq N(I_i)$.

As mentioned earlier in sections 3 and 4, the query can be submitted at any arbitrary summary-schemas node. In particular, when a K-nearest-neighbor query is submitted to the summary-schemas model, the searching is restricted within a small region rather than the whole indexing hierarchy. Assume the nearest neighbors are ordered by their similarities as $I'_1, I'_2, \ldots, I'_K$, the searching of I'_2 will be restricted within an area near the place of I'_1, which makes $W^*(I'_2) \leq N^*(I'_2)$. Hence,

$$\overline{W}^* = \sum_{i=1}^{n} [P(I_i)\, W^*(I'_i)] \leq \sum_{i=1}^{n} [P(I_i)\, N^*(I'_i)] \qquad (4)$$

Considering equation (3) and *Proposition 1*, we obtain

$$\overline{W}^* \leq \sum_{i=1}^{n} [P(I_i)\, N^*(I'_i)] \leq \sum_{i=1}^{n} [P(I_i)\, N(I'_i)] \leq \overline{W} \qquad (5)$$

Hence, the UCSM achieves the optimal performance in terms of searching cost.

6 Further Discussions

In addition to the performance consideration, another important factor – imprecise query processing – favors the choice of summary-schemas model as the underlying platform for content-based indexing. Most of the previous researches [11-18] in content-based retrieval focus on searching cost and similarity comparisons, and do not consider the imprecise query processing. As mentioned earlier, the summary-schemas hierarchy contains two types of summary schemas: the lower-level summary schemas generated by optimization of features, and the higher-level summary schemas constructed from content abstraction of lower-level summary schemas. The higher-level summary schemas may reveal some semantic content beyond the features extracted from the underlying data objects. For example, an image containing "flowers" and "smiling faces" may express the concept of "happiness". For simplicity, we denote the lower-level summary schemas as "quantitative summaries", and denote the higher-level summary schemas as "descriptive summaries".

Section 4 presented an optimization algorithm for generating quantitative summaries. However, these quantitative summaries may not be able to reveal the implication of image objects. For instances, gestures, facial

expressions, and background settings may have some implications that can only be extracted with human senses. Fortunately, these implications can be integrated within the higher-level summary schemas (descriptive summaries).

The descriptive summaries obtain the implications with the help of some common-sense rules, which indicate the semantic relationships between visual components and their symbolic meanings. For instance, "sun + flowers + smile" means "happiness", and "white doves + olive" symbolizes "peace". Some complex image objects may generate multi-level descriptive summaries.

With descriptive summaries, an imprecise query can be processed as follows: First, find a summary schema that matches with the query; then decompose the imprecise query into simpler descriptive summaries (or quantitative summaries) as sub queries; and finally combine the results from the decomposed sub queries. The capability of processing imprecise queries drastically enhances the searching power of the UCSM-based search engine, and makes the UCSM distinguished from other content-based indexing models.

7 Conclusions

We proposed a novel content-aware retrieval model for image data objects in heterogeneous distributed database environment. In contrast with the traditional feature-based indexing models, the proposed model employs a concise descriptive term – ubiquitous content summary – to represent the semantic contents of image objects. In short, the proposed model offers the following advantages: (1) the concise summary accurately represents the semantic contents of image objects using optimized logic terms; (2) the descriptive summary enables the search engine with capability of handling imprecise queries; and (3) the performance of content-based indexing within the UCSM hierarchy is optimal in terms of searching cost. Our future work would include improvements of the UCSM prototype, such as more efficient summarization strategies and adaptation to wireless network environments.

References

[1] A. R. Hurson and M. W. Bright. Multidatabase Systems: An Advanced Concept in Handling Distributed Data. 1991, Advances in Computers 32: 149-200.

[2] W. I. Grosky. Managing image information in database systems. Communications of the ACM, 1997, 40 (12): 73-80.

[3] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yahker. Query by image and video content: The QBIC system. IEEE Computer, 1995, 28 (9): 23-32.

[4] J. Kominek and R. Kazman. Accessing image through concept clustering. Proceedings of the SIGCHI conference on human factors in computing systems, 1997: 19-26.

[5] A. P. Pentland, B. Moghaddam, T. Starner, M. Turk. View-based and Modular Eigenspaces for Face Recognition, IEEE Conf. on Computer Vision and Pattern Recognition, 1994: 84-91.

[6] G. Auffret, J. Foote, C. Li, B. Shahraray, T. Syeda-Mahmood, and H. Zhang. Image access and retrieval (panel session): The state of the art and future directions, Proceedings of the seventh ACM international conference on image, 1999, ACM Image (1): 443-445.

[7] J. B. Kim and H. J. Kim. Unsupervised moving object segmentation and recognition using clustering and a neural network. Proceedings of the 2002 International Joint Conference on neural networks (IJCNN '02), 2002, 2: 1240-1245.

[8] C. Carson, S. Belongie, H. Greenspan and J. Malik. Region-based image querying. Proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries, 1997: 42 -49.

[9] A. B. Benitez. Semantic knowledge construction from annotated image collections. Image and Expo, 2002. 2: 205-208.

[10] M. Simard, S. S. Saatchi and G. DeGrandi. The use of decision tree and multiscale texture for classification of JERS-1 SAR data over tropical forest. IEEE Transactions on Geoscience and Remote Sensing, 2000, 38 (5): 2310–2321.

[11] H. Samet. The Quadtree and related hierarchical data structures. ACM computing surveys, 1984, 16 (2): 187-260.

[12] A. Guttman. R-trees: A dynamic index structure for spatial searching. ACM SIGMOD international conference on Management of data, 1984.

[13] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles, Proceedings of the ACM SIGMOD international conference on Management of data, 1990, 19 (2): 322-331.

[14] N. Katayama, S. Satoh. The SR-tree: An Index Structure for High-dimensional Nearest Neighbor Queries, Proceedings of the ACM SIGMOD international conference on Management of data, 1997, 26 (2): 369-380.

[15] J. T. Robinson. Physical Storage Structures: The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes. Proceedings of the ACM SIGMOD international conference on Management of data, 1981: 10-18.

[16] A. W. Fu, P. M. Chan, Y. Cheung, and Y. S. Moon. Dynamic VP-tree indexing for n-nearest neighbor search given pair-wise distances. VLDB Journal, 2000, 9(2): 154–173.

[17] W. Hsu, T. S. Chua, and H. K. Pung. Approximating content-based object-level image retrieval, Image Tools and Applications, 2000, 12 (1): 59-79.

[18] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments,, IEEE Transactions on Image Processing, 2000, 9 (1): 20–37.

[19] M. W. Bright, A. R. Hurson, and S. Pakzad. A Taxonomy and Current Issues in Multidatabase Systems. IEEE Computer, 1992, 25 (3): 50–60.

[20] M. R. Rezaee, P. M. J. van der Zwet, B. P. E. Lelieveldt, R. J. van der Geest, and J. H. C. Reiber. A Multiresolution Image Segmentation Technique Based on Pyramidal Segmentation and Fuzzy Clustering. IEEE Transactions on Image Processing, 2000, 9 (7): 1238-248.

[21] B. Heisele and W. Ritter. Segmentation of Range and Intensity Image Sequences by Clustering. 1999 International Conference on Information Intelligence and Systems, 1999: 223–225.

[22] D. Yu and A. Zhang. Clustertree: Integration of Cluster Representation and Nearest Neighbor Search for Image Databases. 2000 IEEE International Conference on Image and Expo, 2000, 3: 1713-1716.

[23] Y. Konig and N. Morgan. Supervised and unsupervised clustering of the speaker space for connectionist speech recognition. 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-93), 1993, 1: 545-548.

[24] Jia Wang, W. Yang and R. Acharya. Color clustering techniques for color-content-based image retrieval from image databases. IEEE International Conference on Image Computing and Systems, 1997: 442–449.

[25] J. Malik, D. A. Forsyth, M. M. Fleck, H. Greenspan, T. Leung, C. Carson, S. Belongie, and C. Bregler. Finding objects in image databases by grouping. International Conference on Image Processing, 1996, 2: 761-764.

[26] S. D. MacArthur, C. E. Brodley, and C. Shyu. Relevance feedback decision trees in content-based image retrieval. IEEE Workshop on Content-based Access of Image and Video Libraries, 2000: 68 –72.

[27] I. K. Park, etc. Perceptual grouping of 3D features in aerial image using decision tree classifier. 1999 International Conference on Image Processing (ICIP 99), 1: 31-35.

[28] A. Namasivayam. The use of fuzzy rules in classification of normal human brain tissues. The Third International Symposium on Uncertainty Modeling and Analysis and Annual Conference of the North American Fuzzy Information Processing Society (ISUMA - NAFIPS '95), 1995: 157 -162.

A Gossip-based Distributed News Service for Wireless Mesh Networks

Daniela Gavidia
Faculty of Science
Department of Computer Science
Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV
Amsterdam, The Netherlands
Email: daniela@cs.vu.nl

Spyros Voulgaris
Faculty of Science
Department of Computer Science
Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV
Amsterdam, The Netherlands
Email: spyros@cs.vu.nl

Maarten van Steen
Faculty of Science
Department of Computer Science
Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV
Amsterdam, The Netherlands
Email: steen@cs.vu.nl

***Abstract*— The prospect of having an easily-deployable, self-configuring network for a relatively low investment has made wireless mesh networks an attractive platform to provide wireless services. With the significant attention currently placed on wireless mesh networking, deployment of these mesh networks may be imminent. However, even with the infrastructure in place, development of flexible middleware has yet to reach a level where it can deliver on the promise of mesh networks due to the special characteristics of wireless connectivity. In this paper, we propose a fully decentralized news service based on epidemics. Its simple design makes for a scalable and robust solution, flexible enough to be used as the basis for other more sophisticated applications.**

I. Introduction

As advances in wireless networking continue, we are gradually seeing a shift in which distributed (middleware) systems are moving from wired networks to heterogenous or completely wireless systems. Notably, wireless mesh networks [1] offer the facilities to quickly and cheaply set up a networking infrastructure that can easily span the size of a city. From a distributed systems perspective, the challenge lies in providing services that can hide the inherent unreliable nature of the underlying infrastructure. This unreliability is caused by failing links and a relatively high rate of joining and leaving nodes (purposefully or unintentionally), which continuously affect the topology of the network.

This instability requires that we seek new solutions to well-known problems. As a step in that direction, we are exploring how gossiping protocols can help in the construction of highly robust services. In this paper, we consider the problem of providing a *news service* that runs entirely on a wireless mesh network. This service provides mobile users news items that are of interest to them. In our approach, we assume that a user, by means of a PDA or a similar small device, can connect to an access point (i.e., router) of a wireless mesh network. When connected, the user can read news items as if accessing a central database where all items are stored. Using content-based filtering, for example by means of SQL-like queries, only the items of interest will be delivered.

The problem we address can best be described as setting up a simple, self-configuring news service in a mesh network, under the condition that it be fully decentralized. The reasons for avoiding a centralized implementation are, in a way, related to the nature of mesh networks. Wireless mesh networks are based on the principle of cooperation between routers, most notably exemplified by routers forwarding packets on behalf of other nodes. With that in mind, we want to steer away from a centralized solution where one node is solely responsible for the availability of the service. A decentralized solution effectively divides the workload (and responsibility) among the collection of nodes providing the service, allowing us to sidestep issues that may arise from having a single point of failure and single ownership of the service. We outline the requirements for a successful implementation of our distributed news service as follows:

- **Ease of deployment** A collection of nodes should be able to start providing the service with minimal configuration. Nodes should be able to join the system without going through complicated bootstrapping mechanisms. In essence, we desire to have a decentralized system where nodes can start making a contribution to the service as soon as they are operational.
- **Minimal requirements** Contributing to the service should not be a burden to the nodes in the mesh network. Memory and computational requirements should be small enough to allow any router to be part of the service. No powerful nodes are expected to be in place for high-performance tasks.
- **Robustness** The system should be minimally affected by nodes joining and leaving the network. Moreover, recovery from significant changes in membership should be prompt.
- **Scalability** The service should be able to perform adequately in the face of increasing number of nodes and news items being published.
- **Effectiveness** When an item is published, it should be made available to the interested users in a timely manner.

We expect to meet these requirements by having the routers in the mesh backbone exchange news items using the epidemic protocol we introduce in Section III. Epidemic (or gossip-

based) techniques have proved to be a robust, efficient, and scalable solution for disseminating information in peer-to-peer networks [2], [3], [4]. Aside from the robustness and scalability inherent to gossiping, the protocol we present is characterized by simple, independent one-to-one interactions. The simplicity of this approach allows any router willing to participate in the service to start contributing as soon as they come in contact with a router that is already providing the service. By basing our solution on gossiping, we expect to be less vulnerable to topology changes.

Our main contribution is that we embrace the unpredictable nature of wireless networks and attempt to use this to our advantage by implementing a gossip-based solution. Our approach skews deterministic routing in favor of probabilistic delivery of news. As a result, we can deliver a scalable and robust service with predictable behavior for large-scale deployments.

The remainder of this paper is organized as follows. In the next section, we describe our system model, specifying the assumptions we make and describing an application scenario for the news service. Section III details the implementation of the gossip protocol executed by the network of mesh routers. In Section IV, the performance of the service from the user's perspective is analyzed. Related work is discussed in Section V. Section VI presents a discussion followed by conclusions and final remarks in the last section.

II. System Model

The service we propose is provided by a mesh backbone composed of a large number of wireless routers. Users running the news service are able to publish events, which we call *news items*, of interest to other users. These users carry around *clients*, which are portable devices capable of connecting to the mesh backbone to retrieve news items. Essentially, the clients poll the routers for news items matching the interests of users. By specifying their preferences in advance and using them for filtering, users are able to receive in their portable devices only relevant news items.

When initially contacting a mesh router, clients are expected to send a filter to be used to identify the items of interest to the user. As long as the client maintains a connection to the router, it will receive updates whenever new items that match the users interests are received. Filtering is done at the router to avoid excessive communication with the client devices, which may have limited power supplies. Filters are not propagated through the network.

A. Assumptions

We assume the presence of a large collection of mesh routers forming a mesh backbone. These mesh routers are not mobile and, as a whole, provide coverage for an extensive geographical area. As part of the fixed infrastructure, they do not have strict constraints on power consumption. We expect these routers to have a dedicated amount of memory space to be used for storing news items. These caches will be updated periodically using the gossip protocol described in Section III.

News items are propagated through the network in the form of *news entries*. While a news item is a piece of information, a news entry is the representation of the news item in the network and for each news item several news entries may exist. The dissemination of news entries is done primarily within the mesh backbone. Each router can communicate wirelessly with the routers within its range. These routers are called its *neighbors*. A unique *id* is associated with each router. The entries that a router inserts into the network can be uniquely identified by a combination of the router id and a sequence number. In its most basic form, a news entry contains a unique id, a timestamp and a time-to-live. There may be other fields of information depending on the application. A limited number of these entries can be stored by each router in its local *cache*. In our experiments, the size of the cache is defined by the parameter c, which is the same for all routers. The storage capacity of the network as a whole is then $N \times c$, where N is the number of routers in the network. Routers in the network gossip periodically, exchanging the entries in their caches. We define a *round* as a gossiping interval in which each router initiates an exchange once.

The clients in our system are, for the most part, portable devices, such as phones, laptops or PDAs. These devices have limited power supplies and, for that reason, do not participate actively in the dissemination of news items. They do, however, engage in communication with the routers to be updated on news events.

B. Application Description

To illustrate the usefulness of the service, we propose a possible application scenario: advertising in a shopping center where products on sale need to be promoted. In this scenario, routers could be located at any other shop. Some routers may already be in place for use as hotspots or as part of a store's accounting system. As computers have become prevalent in business environments, we do not expect lack of infrastructure to be a major obstacle for the deployment of the mesh network. With the mesh network in place, news items advertising products would be disseminated through the mesh network and be picked up by the mobile devices that costumers carry.

News entries have a limited lifetime. After this time period expires, the information they carry is no longer valuable to clients and should be flushed from the network. Going back to our example, the lifetime of entries could relate to the time period when a sale is effective (for example, drink at a discount price during lunch time).

At any point in time, a router will have a partial view of the complete set of news items in its cache. We do not expect each router to store all items. Instead, each router will devote a fixed amount of memory to store entries it discovers through communication with other routers. Periodically, this view will be refreshed with different news entries. According to the interests that costumers have expressed when contacting a router, their mobile clients will be updated with relevant advertisements.

```
/*** Active thread ***/
// Runs periodically every T time units

Q = selectPeer()
buff_send = selectItemsToSend()
send buff_send to Q
receive buff_recv from Q
cache = selectItemsToKeep()
```

(a)

```
/*** Passive thread ***/
// Runs when contacted by another router

receive buff_recv from any P
buff_send = selectItemsToSend()
send buff_send to P
cache = selectItemsToKeep()
```

(b)

Fig. 1. Skeleton of an epidemic protocol.

III. Shuffle Protocol

When a router participates in a gossip exchange, it assumes either an *active* or a *passive* role. Each router initiates an exchange once per round. We refer to the router that initiates the exchange as the active one, while the one that is contacted assumes the passive role.

The data exchange between routers follows a predefined structure. Figure 1 shows the skeleton of the push-pull epidemic protocol we use for communication within the mesh backbone. Three methods, `selectPeer()`, `selectItemsToSend()` and `selectItemsToKeep()` represent the core of the protocol. By implementing different policies in these methods, various epidemic protocols, each with its own distinctive characteristics, can be instantiated.

Based on the structure shown in Figure 1, we introduce an epidemic protocol we call *shuffle*. The shuffle protocol is characterized by avoiding the loss of data during an exchange. It achieves this by establishing an agreement between peers that each peer will keep the entries received from the other after the exchange takes place. We will elaborate on the details of the exchange later on.

The shuffle protocol is partly based on a peer-to-peer protocol used for handling flash crowds [5], which we recently enhanced in order to maintain unstructured overlays that share important properties with random graphs [6]. The most important observation to make is that any two nodes that engage in a shuffle essentially *swap* a number of entries. In doing so, they not only preserve the data that are collectively stored in the network, but also "move" these data around in a seemingly random fashion. The underlying idea is that by randomly shuffling data entries between nodes, all nodes will be able to see all news items eventually.

A. Protocol Policies

In the shuffle protocol, each node agrees to keep the entries received from a neighbor for the next round. This might seem trivial, but given the limited storage space available in each node, keeping the entries received during an exchange implies discarding some entries that the node has in its cache. By picking the entries to be discarded from the ones that have been sent to the neighbor, we ensure the conservation of data in the network. The policies are summarized as follows:

Method	Description
`selectPeer()`	Select a neighbor randomly
`selectItemsToSend()`	Randomly select s entries from the local cache. Send a copy of those entries to the selected peer.
`selectItemsToKeep()`	Add received entries to the local cache. Remove repeated items. If the number of entries exceeds c, remove entries among the ones that were previously sent until the cache contains c entries.

B. Simulation Setup

In order to observe the behavior of the protocol in large-scale settings, a series of simulations were conducted. We have learned from earlier studies of other epidemic protocols [7] that the results from emulations running in a cluster of hundreds of nodes yield strikingly similar results to simulation results when observing large-scale behavior. For this reason, we decided to study the behavior of the protocol presented in this paper through extensive simulations. The results presented in this section correspond to a network of 10000 nodes with a cache size of c, which may vary in different experiments. Two types of topologies were used in the experiments:

- *Grid topology* The nodes were set up in a square grid topology, with 100 nodes on each side over an area of 100×100 units. Two cases were explored: (a) the range of each node was set to 1 unit, making communication possible with the node's immediate neighbors to the North, South, East and West. On average, each node had 3.96 neighbors (due to the effect of boundary nodes with less than 4 neighbors); (b) the range of each node was set to 2 units, making communication with 12 immediate neighbors.
- *Random topology* The nodes were placed randomly in a square area of 100×100 units. Nodes were allowed to reach neighbors within a range of 2 units, which was enough to guarantee that each node had at least one neighbor and that a path between any two nodes existed. The average number of neighbors for each node was 12.19.

Both topologies were used to study the behavior of the protocols. The experiments that we conducted focused on two characteristics observed during the execution of each epidemic

protocol (a) the replication of items in the network and (b) the time required to reach all the nodes in the network.

C. Properties

To understand the behavior of the protocol, we focus on the way a single news entry traverses the network. On first instance, a news entry is inserted into the network by a router. Subsequently, the entry takes a step (moves to the cache of another router) whenever the router that currently holds the entry participates in an exchange. For every execution of the protocol, the next step of the entry is chosen randomly. As a consequence, the path followed by an individual entry consists of a series of random steps. This behavior is analogous to a random walk in the space defined by the mesh network.

Additionally, as an entry moves from router to router, there is a chance that it will be replicated in the caches of the routers it has passed through, given that there was space available. It follows that a news item may have several news entries in the network at the same time. For that reason, when referring to an item in the network we are actually referring to all news entries that represent that news item. These entries have the same id. In the next sections we study the way these entries are replicated through the network.

1) Distribution of Storage Capacity: Let us first consider how different news items are distributed through the network. After running the protocol for several rounds, we observe that the storage capacity of the network is evenly divided between the different items. By this, we mean that the slots available to store news entries are used in a balanced way, with each news item being able to place approximately the same number of entries in the network. This behavior is not programmed into the algorithm, but it is an emerging property resulting from its repeated execution.

The value to which the number of entries of an item converges is dictated by the number of different news items in the network. Given a network of size N where all nodes have a cache size of c, the network has a total capacity of $N \times c$. These $N \times c$ available slots have to be filled with d different news items. Because of the randomness introduced when choosing which entries to exchange, the total capacity should eventually be evenly divided between the different items resulting in an average of $\frac{N \times c}{d}$ entries for each of the d news items. Considering that the protocol does not allow more than one news entry representing the same news item in the same cache, this means that c/d of the nodes should have an item of each of the d different ids:

$$\#\text{ entries per item} = \frac{\text{capacity of the network}}{\text{number of news items}} = \frac{N \times c}{d}$$

Figure 2 shows the convergent behavior of the protocol. For the experiment, a collection of 10000 nodes were placed in a grid topology with 4 neighbors per node and 10 nodes were randomly selected to generate different news items. Time is measured in *rounds*, where a round is a gossiping interval in which each node executes the exchange protocol once. After an initial stabilization period, the number of entries in the system for each of the 10 items converges to the same value.

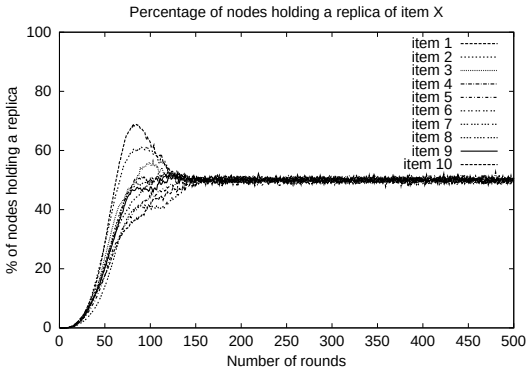


Fig. 2. Convergent behavior illustrated by having 10 nodes that generate news items in a network of 10000 nodes.

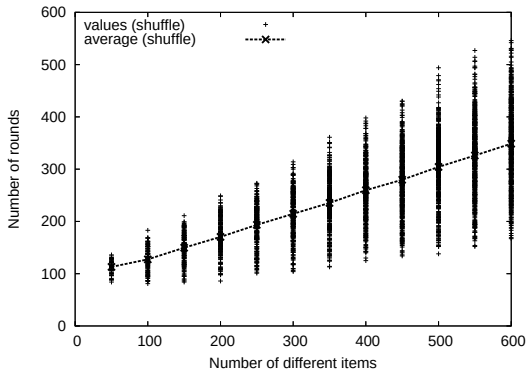


Fig. 3. Number of rounds required for all the routers in the backbone to have seen an item using shuffling on a grid topology.

According to our previous reasoning, this value should be $10000 \times 5/10 = 5000$, meaning that 50% of the nodes in the network have an entry from one of the 10 different news items available, which is confirmed by our experiments. Similar convergent behavior was observed when experimenting with other topologies.

2) Dissemination Speed as a Function of the Diversity of News Items: To demonstrate the effectiveness of shuffling for disseminating information, we have conducted experiments that show the effect of the number of different items on the dissemination speed of the items through the network. In this section, we look at the time needed for the news items to have reached all routers in the network. The results presented here correspond to a mesh backbone of 10000 routers. Unless explicitly stated, the routers were set up in a rectangular grid topology, with 100 routers on each side. For the experiments, we measure the time it takes for the items to reach all the routers in the network.

Figure 3 shows the time, measured in rounds, required for various different items to have passed through the caches of all the routers in the backbone. The cache size for all routers was set to 50 and all items in the cache were shuffled in each round. In each experiment, a different number of distinct items (starting at 50 and up to 600, with increments of 50) were inserted into the backbone by routers located in

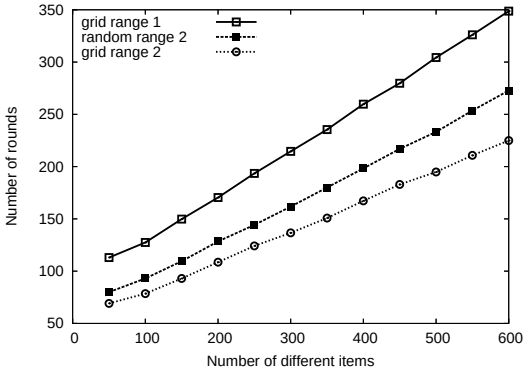


Fig. 4. Number of rounds required for all the routers in the backbone to have seen an item using shuffling on three different topologies.

random locations. For each news item, the time required to traverse the backbone was measured. Due to randomness in the exchanges and the location of the routers inserting the news items, the time measured for an individual news item may vary significantly compared to the measurements for other items. By calculating the average time for a news item to go through the mesh backbone, we can observe that as the diversity of news items in the network increases the average time for a specific news item to reach all routers increases linearly. We observe that this linear behavior is maintained when conducting the experiments with different topologies, as shown in Figure 4.

In our third set of experiments, we focus on the effect of the cache size on the dissemination speed. As before, we look at the average time required for an item to have reached all routers in relation to the number of different items being gossiped. The results, shown in Figure 5, reveal that the slope of the curve of average values is directly related to the number of items being shuffled. There is an inversely proportional relationship between the number of items being exchanged and the slope of the curve. The four curves shown correspond to experiments with a cache size of 30, 40, 50 and 60 items. In all cases, all entries in the cache were exchanged. By doubling the number of entries shuffled from 30 to 60, the average time for news items to pass through every router in the backbone is virtually divided in half. Such a characteristic, as well as the predictable behavior with an increasing number of different items, are important factors to consider when choosing an appropriate value for the cache size c and the number of entries to shuffle.

3) Robustness: In order to test the robustness of our system in the case of node failures, we look at a scenario where the nodes within a limited area go down, not unlike what would happen in case of a power outage. The experiment, performed with 10000 routers arranged in a grid with range 1, consists of observing the number of entries per item in the mesh backbone before, during and after the failure of all routers within a square area. We assume that when a router fails, all the entries in its cache are lost. When the router goes up again, its cache is empty and has to be populated again.

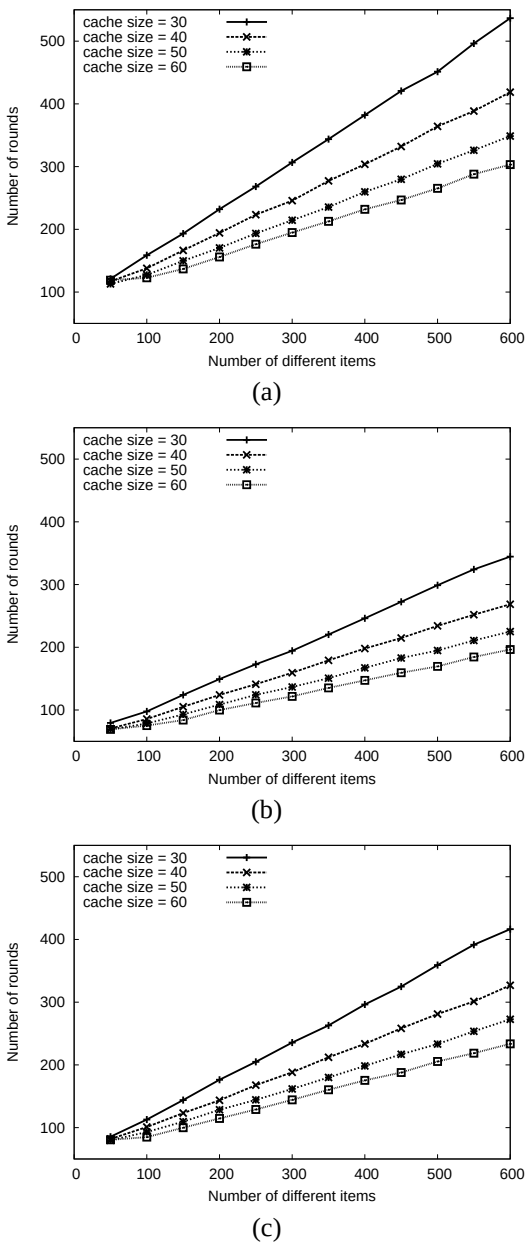


Fig. 5. Number of rounds required for all the nodes in the network to have seen an item for different cache sizes with (a) grid topology (range 1), (b) grid topology (range 2) and (c) random topology (range 2). All entries in the cache are exchanged.

Figure 6 shows the results of the experiment when 49% of the routers experience a failure at the same time and recover 100 rounds later. The routers chosen for failure were arranged in a 70×70 square inside the 100×100 grid. Like the experiment in Figure 2, 10 items are being shuffled in the network and all routers have a cache size of 5. Once the number of entries per news item has converge to the same value, the routers chosen for failure go down. As could be expected, given that the entries were randomly located throughout the network, the number of entries per item is virtually cut by half once the failures occur. When the routers that failed rejoin

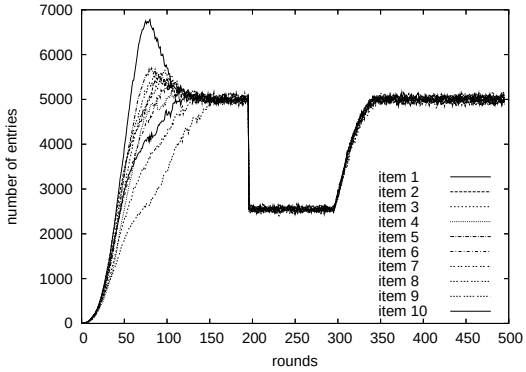


Fig. 6. Number of entries per news item. 49% of the routers go down at round 200 and recover at round 300.

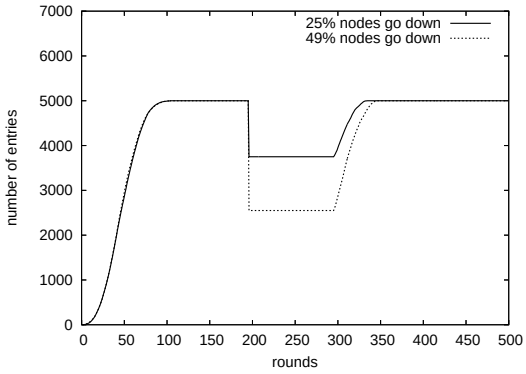


Fig. 7. Average number of entries per item. Routers go down at round 200 and recover at round 300.

the network, we observe a smooth transition to the previous state, with entries quickly populating their caches. Unlike the first rounds, the entries per item are replicated at roughly the same rate. This is due to the fact that the routers surrounding the area of failure already have their caches full of entries and can update the routers that failed as soon as they become operational again.

For a clear view of the recovery time, Figure 7 shows the average number of entries per item. In addition to the experiment presented earlier, we include the case where 25% of the routers fail. These routers are arranged in a 50×50 square. Comparing both curves, we observe the same behavior up to the moment of node failures. At that point, the average number of entries per items falls according to the loss of storage space. The recovery in both cases is quick despite the difference in the number of routers that failed.

The speedy recovery of the affected area can be attributed to information flowing in from multiple sides. For an affected square area, we would expect the recovery time to be proportional to the square root of n, where n is the number of routers that experience a failure. As can be seen in Figure 8, this seems to be the case. The results shown in the figure were obtained using the random topology with range 2. Several experiments where routers within a square area failed were conducted. For each experiment, a square area of a different

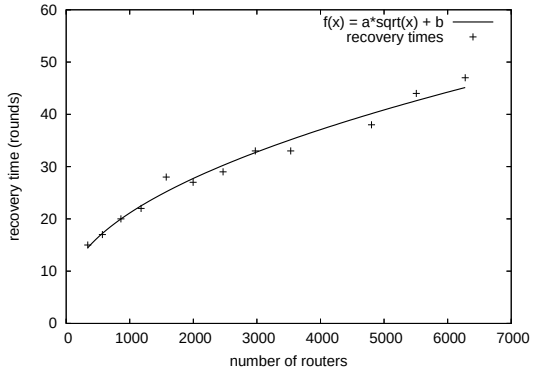


Fig. 8. Recovery times when an increasing number of routers fail.

size was used, ranging from 20×20 up to 80×80 with increments of 5 distance units on each side. Figure 8 shows that, indeed, the recovery times obtained from the simulations tend to be proportional to the square root of the number of routers affected. To verify this, we also plot the curve a*sqrt(n)+b which was obtained through linear regression. The constants have values 0.504244, and 5.18383, respectively.

IV. Performance on the Client Side

In this section, we take a look at the performance of the news service from the user's perspective. Users of the service have access to news items through *clients* in the mesh network. These include portable devices such as laptops, PDAs or other hand-held devices. Due to the variety of news items available in the network, users need to configure their clients to retrieve items matching the users interests. Item retrieval is based on content filtering. Once a connection to a router is established, clients must submit filtering criteria for the router to identify which news items to forward to the client.

A. Recall Rate

To evaluate the effectiveness of the news service from the user's perspective, we observe the *recall rate* of items over time. The recall rate is defined as the number of items of interest to a user that a router delivers to the client device over a period of time versus the total number of relevant items. We test the recall rate through the following experiment:

- A network of 2500 routers is arranged in a 50×50 grid. Each router can communicate with its neighbors to the North, South, East and West.
- 50 users are positioned at random locations.
- 500 news items are being shuffled in the network.
- The interests of the users match 100 news items.

The news items are published at random locations in the network and shuffled until the number of entries for each item converges to roughly the same value (as seen in Section III-C.1). At that point, the clients connect to the nearest router expressing interest in certain kinds of items. A router responds by forwarding the matching news items seen in its cache to the client. Caches are updated with every gossip round prompting the delivery of previously undiscovered items to the client.

As a result, we expect the recall rate to increase as the client spends more time connected to a router.

The results of repeating the experiment with different cache sizes can be seen in Figure 9. The figure shows the average recall rates for the 50 users. In all cases, the recall rates increase rapidly during the initial rounds and slow down when most items have already been discovered. As could be expected, larger caches lead to higher recall rates of items. This is due to a higher storage capacity in the network that allows for more entries to be placed for each news item. Therefore, the probability of finding a particular item in the cache of a router increases.

It should be noted that an increase in the total number of news items would slow down the recall rate, as dissemination speed decreases with the number of news items in the network. This effect can be countered by an increase in cache size. In the remainder of this section, we take a fixed number (500) of news items and explore the effect of modifying other parameters, such as the cache size, the number of items shuffled and the topology of the network, on the recall rate.

B. Probability of seeing an item

Executing the shuffle protocol until the storage capacity of the network is full yields a probability of c/d of finding a particular item when examining the cache of a router picked at random, for $c \leq d$. If we define the success of our experiment as finding a particular item in a random cache and knowing that the probability of success remains constant, the probability of succeeding after performing the experiment $k \geq 1$ times is:

$$p(k, c, d) = 1 - \prod_{i=1}^{k} (1 - prob\_success(i)) = 1 - \left(1 - \frac{c}{d}\right)^k$$

Figure 10 shows the probability of finding an item in a cache selected at random after k attempts for different cache sizes. We observe a similar, although not identical, behavior to the recall rate results presented in the previous section. This is not surprising, as the shuffle protocol ensures that after each round a router refills its cache with entries received from a neighbor chosen at random. However, due to the locality of the gossip exchanges, when looking at the cache of the same router for several rounds, we are bound to discover the items held by our neighbors first. This limits the variety of items we might see as our neighbors are more likely to hold many of the same items as we do in comparison to a randomly chosen router in the network. This accounts for the slightly lower recall rate in the experimental results compared to the probability of seeing an item when selecting a random cache every time.

C. Improving Recall Rate

Shuffling provides a random sample of the collection of items in the network at every round for each router, however, as can be inferred from Figure 9 and 10, there is a correlation in the items seen from one round to the next, which accounts for less than optimal recall rates. In other words, the reason for the recall rate results not being identical to the probability in Figure 10 is due to the results from each round not being

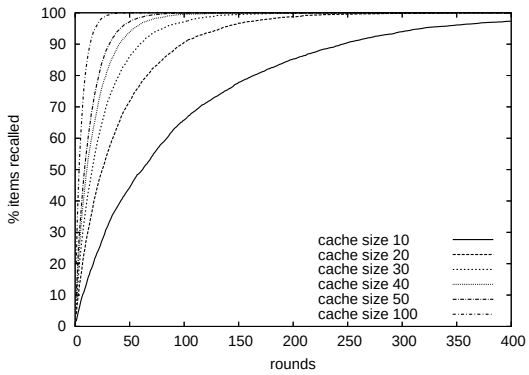


Fig. 9. Recall rate of news items. All entries in the cache are exchanged.

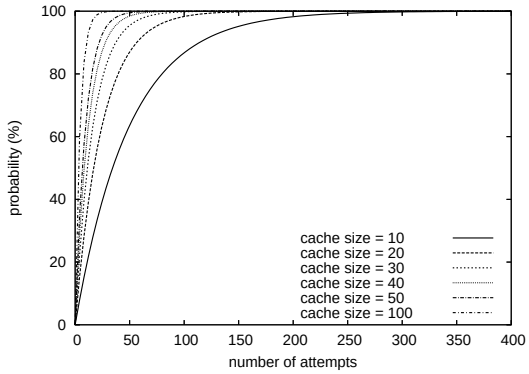


Fig. 10. Probability of recalling an item from a randomly filled cache.

independent. This can be attributed to the lack of variety of items in the neighborhood of a router.

Figure 11 shows the effect of the neighborhood in the recall rate. We confirm that the recall rate was being hampered by each router having a limited neighborhood by showing that the probability of seeing an item when a cache is picked randomly is the same as the recall rate when the range of a router is such that it can reach any other router in the network. In the graph, the results for a network of routers with range 100 and $p(k, 50, 500)$ overlap. We also show the impact in performance of doubling the range from 1 to 2 units, effectively increasing the number of neighbors from 4 to 12. This experiment shows that it is not necessary to be able to reach every node in the network to achieve a close-to-optimal recall rate. Finally, as a worse case scenario, we show what happens if the routers are arranged in a single line, where each router can only reach its neighbors to the left and right. In this case, the recall of items after the first few rounds becomes increasingly slow. We attribute this to new items being hard to come by after the items of interest in the immediate neighborhood have been discovered. Having only two neighbors, the likelihood of new items reaching the neighborhood is reduced, requiring more iterations of the protocol to update a cache with different items. Obviously, this topology is not realistic and should be avoided.

Another option for improving the recall rate without increasing the amount of entries exchanged per round is to

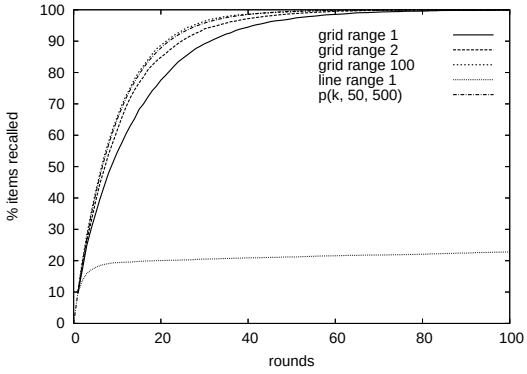


Fig. 11. Recall rate of news items.

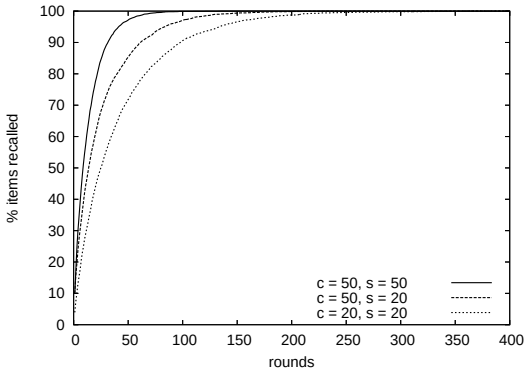


Fig. 12. Recall rate of news items.

increase the cache size. The results presented previously assumed that all entries in the cache were exchanged in each round. Figure 12 shows how increasing the cache size while exchanging the same amount of entries provides an initial boost in the recall rate. However, since the speed at which news items move through the network depends on the number of entries exchanged per round, having a bigger cache does not provide any benefits for finding the last few items that were not originally in the vicinity of the router.

V. Related Work

From a functional point of view, the news service we propose has some similarity with content-based distributed publish/subscribe systems [8], [9], [10], [11]. With the increase in popularity of wireless technology, some publish/subscribe systems have been extended to support for mobile, wireless clients [12]. For the most part, the systems proposed use a single tree-shaped overlay to interconnect a set of brokers which cooperate to deliver the events published to the appropriate subscribers. This approach, while efficient under static conditions, might face robustness and scalability issues in highly dynamic environments, such as wireless networks with mobile users. Efforts in maintaining a tree overlay under frequent topology changes aim at dealing with these issues [13]. However, depending on how frequently the changes occur, maintaining a tree may introduce additional overhead and complexity. Our approach offers robustness and scalability at the cost of periodic communication for gossiping.

Content-based publish/subscribe projects aimed explicitly at wireless, mobile environments [14], [15], [16] are more closely related to our work. In particular, systems that rely on probabilistic techniques for the delivery of events. In particular, [17] combines deterministic routing with probabilistic techniques to increase the resilience when faced with topology changes.

Our work also relates, in a way, to efforts in distributed storage [18], [19]. Like our news service, these systems rely on data redundancy to ensure robustness when node failures occur. However, while most of these systems carefully place replicas based on the reliability of nodes, we replicate items and relocate them in a random fashion. We do, nevertheless, manage to use the storage capacity in a fair manner, dynamically adjusting the number of replicas of an item according to the number of items in the network.

VI. Discussion

As mentioned in the introduction, one of the main advantages of having a decentralized system like the one we propose is avoiding single ownership of the service. Single ownership implies that one entity is fully responsible for the availability and quality of the service. This may not be a bad thing from the point of view of managing the system, however it restricts others from contributing to the service even when resources are available. One of the strenghts of our news service is the flexibility it allows for routers to have some control over the quality of service they offer. As explained in Section IV, the quality of the service as perceived by the users can be improved by increasing the wireless range or the amount of memory allocated for storing entries. Decisions to do so can then be taken on an individual basis by the administrators of each router.

Taking a more active approach for the recall of items is also a possible way of improving the perceived performance from the users point of view. As explained, clients connect to a nearby router and retrieve news items matching the user's interests. While the matching news items from the router's cache will be made available to the client immediately, discovering the totality of relevant news items may take several rounds and, depending on the time period between rounds, this delay might inconvenience some users. Instead of passively waiting for the news items to arrive, a router may decide to forward the user's filter to other routers, thus increasing the chances of discovering relevant news items. For example, we can calculate that for $c/d = 0.2$ it would take at least 10 rounds to retrieve approximately 90% of all items. In this case, by forwarding the filter to 4 other routers, the client could receive almost all news items in 2 rounds.

The flexibility of being able to independently decide on the amount of resources to invest in the news service coupled with the minimal requirements to participate opens up the possibility of deploying the service on a large scale using heterogeneous nodes. Deploying the service over large geographical areas, such as a campus or a city, may require

some considerations in the dissemination of news items. As mentioned before, we impose a time limit for the validity of the entries in the network. However, when disseminating the entries over a large area, it may also be necessary to establish geographic constraints. By adding location-awareness to the shuffling of entries, news items could be dispersed over limited areas. For example, an entry may be forwarded only within a radius from the location where it originated. As a result of restricting the area over which an entry can travel, space which would otherwise be taken by these entries is freed increasing the number of different items that the network can hold.

Another consideration to keep in mind is security. A gossip-based system like the one we propose is specially susceptible to denial-of-service attacks. We can imagine a scenario where a malicious node generates bogus news items and inserts them into the network at a high rate. Having large numbers of items at the same time slows down the dissemination speed, as there are less entries per item in the network. Without any security mechanisms in place, a single node could virtually bring the service to a halt. It is clear that some kind of regulation regarding who can publish news items is necessary.

VII. Conclusions and Future Work

In this paper, we presented a highly robust distributed news service suitable for wireless mesh networks. We have shown that by using an epidemic protocol at the core of our service, we can provide an efficient and scalable solution for delivering news items while, at the same time, offering the participating routers the flexibility of managing their own resources to better suit their clients needs. Through the use of simulations, we analyzed the effectiveness and robustness of the dissemination of items through the mesh backbone. We corroborated the effectiveness of the service by taking the user's perspective and providing an analysis of the quality of the service in terms of the delivery of relevant news items to the client devices.

Regarding future work, in addition to the issues already addressed in the discussion, we expect to explore other types of services that could be deployed on top of these networks. Even though large deployment of mesh networks is not yet a common ocurrence, we intend to focus on developing distributed solutions for large-scale networks with the belief that a collaborative effort can yield efficient results under the sometimes unpredictable conditions of wireless networks.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," *Computer Networks Journal (Elsevier)*, March 2005.

[2] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," in *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing (PODC '87)*, ACM, Ed., Vancouver, Canada, August 1987, pp. 1–12.

[3] K. P. Birman, "The surprising power of epidemic communication," in *Future Directions in Distributed Computing: Research and Position Papers*, January 2003, vol. 2584/2003, pp. 97–102.

[4] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen., "The Peer Sampling Service: Experimental Evaluation of Unstructured Gossip-Based Implementations," in *Middleware 2004*, ser. Lecture Notes on Computer Science, vol. 3231, ACM/IFIP/USENIX. Berlin: Springer-Verlag, Oct. 2004, pp. 79–98.

[5] A. Stavrou, D. Rubenstein, and S. Sahu, "A Lightweight, Robust P2P System to Handle Flash Crowds," *IEEE Journal on Selected Areas in Communication*, vol. 22, no. 1, pp. 6–17, Jan. 2004.

[6] S. Voulgaris, D. Gavidia, and M. van Steen., "Inexpensive Membership Management for Unstructured P2P Overlays," *Journal of Network and Systems Management*, vol. 13, no. 2, pp. 197–217, June 2005.

[7] S. Voulgaris and M. van Steen, "An Epidemic Protocol for Managing Routing Tables in very large Peer-to-Peer Networks," in *Proc. 14th IFIP/IEEE Workshop on Distributed Systems: Operations and Management (DSOM 2003)*, Oct. 2003, pp. 41–54.

[8] A. Carzaniga, D. Rosenblum, and A. Wolf, "Design and evaluation of a wide-area event notification service," *ACM Trans. Comput. Syst.*, vol. 19, no. 3, pp. 332–383, 2001.

[9] G. Cugola, E. D. Nitto, and A. Fuggetta, "The JEDI event-based infrastructure and its application to the development of the opss wfms," *IEEE Trans. Softw. Eng.*, vol. 27, no. 9, pp. 827–850, 2001.

[10] P. R. Pietzuch and J. Bacon, "Hermes: A distributed event-based middleware architecture," in *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society, 2002, pp. 611–618.

[11] B. Segall and D. Arnold, "Elvin has left the building: A publish/subscribe notification service with quenching," in *Proceedings of the Australian UNIX and Open Systems User Group Conference (AUUG'97)*, September 1997, pp. 243–255.

[12] M. Caporuscio, A. Carzaniga, and A. L. Wolf, "Design and evaluation of a support service for mobile, wireless publish/subscribe applications," *IEEE Transactions on Software Engineering*, vol. 29, no. 12, pp. 1059–1071, Dec. 2003. [Online]. Available: http://serl.cs.colorado.edu/ carzanig/papers/

[13] G. Picco, G. Cugola, and A. Murphy, "Efficient content-based event dispatching in the presence of topological reconfigurations," in *Proc. of the 23 Int. Conf. on Distributed Computing Systems (ICDCS 2003)*, 2003. [Online]. Available: citeseer.ist.psu.edu/picco03efficient.html

[14] G. Cugola, A. L. Murphy, and G. P. Picco, "Content-based Publish-subscribe in a Mobile Environment," in *Mobile Middleware*, P. Bellavista and A. Corradi, Eds. CRC Press, 2005, invited contribution. To appear.

[15] Y. Huang and H. Garcia-Molina, "Publish/subscribe tree construction in wireless ad-hoc networks," in *MDM '03: Proceedings of the 4th International Conference on Mobile Data Management*. London, UK: Springer-Verlag, 2003, pp. 122–140.

[16] R. Meier and V. Cahill, "STEAM: Event-based middleware for wireless ad hoc networks," in *Proceedings of the 1st International Workshop on Distributed Event-Based Systems (DEBS '02)*, Vienna, Austria, July 2002.

[17] P. Costa and G. P. Picco, "Semi-probabilistic content-based publish-subscribe," in *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 575–585.

[18] A. Haeberlen, A. Mislove, and P. Druschel, "Glacier: Highly durable, decentralized storage despite massive correlated failuresglacier: Highly durable, decentralized storage despite massive correlated failures," in *Proceedings of the 2ndt USENIX Symposium on Networked Systems Design and Implementation (NSDI '05)*, Boston, Massachusetts, May 2005.

[19] A. Adya, W. J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, and R. P. Wattenhofer, "Farsite: federated, available, and reliable storage for an incompletely trusted environment," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 1–14, 2002.

Message-On-Demand Service in a Decentralized Unified Messaging System

Prem Prakash Jayaraman, Paul Hii, and Arkady Zaslavsky

Abstract— Variety of service integration concepts have emerged during the last few years. Most of these aim at the concept of integrating telecommunication and the data communication technologies. One form of such a system is Unified Messaging. Unified Messaging enables users to manage their messages independent of location, communication device or communication medium. Most of the existing systems provide centralized message-store based access to messages but lack services like user personalization, message notification and are non-pervasive. In this paper, we define UMS as a user-centric system that provides messaging services based on user demands and preferences. We have proposed a decentralized Unified Messaging System (DUMS) that is pervasive and context-aware. Based on our proposed architecture, we successfully implemented and demonstrated a messaging system called *i*-UMS that ensures a user receives almost instantly all messages such as emails, instant messages and so on in an intelligent and most appropriate manner.

***Index Terms*— UMS, Service Adaptation, Pervasive, Service Personalization, Ubiquitous**

I. Introduction

The Internet has now become the universal mode of communication letting users to communicate with each other by a number of means like email, IM, Voice over Internet Protocol (VoIP), etc. This development has led to the convergence of geographically separated users. Number of messaging technologies and services has evolved during the past few decades with the advent of new communication technologies and devices. These messaging systems work in common to deliver messages to the user, but differ in their architectures and message formats. The systems use different communication medium to deliver messages and users' use different end-terminals to receive messages. E.g. SMS is sent and received over GSM network while email is sent and received over a transmission control protocol (TCP). This diversity in the various messaging systems has left end users with a plethora of messaging services each of which provides services in a different format. Also users making use of these messaging services have to consider the recipients' location and end-device to obtain prompt reply from the message recipients [Wj04].This approach is unsatisfactory. Hence, this led to the investigation of a message unification service that enables end-users to send and receive messages irrespective of the messaging system format and architecture. Unified Messaging (UM), which is a widely accepted service, aims at this goal of integrating various messages irrespective of their architecture. Unified Messaging can be defined as a system that enables transfer of messages independent of the underlying message architecture, communication medium and devices. Any form of messages such as email, SMS, fax, voicemail, etc. can be delivered via any communication medium [As99]. The UM can be a powerful means of communication as it seamlessly integrates various communication technologies hence, giving the user the flexibility to receive and send any message, anytime and anywhere [Vs00][As99].

UM incorporates services that convert messages like email, fax, voicemail, SMS from one format to another format. Most of the UMs perform this process of message conversion without taking into consideration the user's preferences. The goal of UM is to deliver messaging services to the user's in the most appropriate method. Current UMs does not entirely fulfill the goals of the unified messaging. Hence message conversion is no longer the primary goal of a UM. The system must be able to adapt itself into the user's environment. It needs to take in account user's context [Dc02], user's end-terminals and user's working environment. This is an intelligent approach that creates the foundation in developing a Unified Messaging System (UMS) that adapts itself based on user's location and user preferences hence enabling the Messaging-On-Demand service. The Message-On-Demand in this context can be defined as a service that provides users with messages on demand based on user's preferences, user's context and user's personalized service preferences. The aim of the user in such an environment is no longer to have ubiquitous access [Vs00] to messages instead personalization of the messaging services based on user's messaging demands. Since the system needs to be context-aware and terminal-aware, the environment in which the devices work can be described as a virtual personal area network (VPAN),

Prem Prakash Jayaraman is a Research Student with Monash University, Melbourne, Australia 3145 (corresponding author phone: +61-3-9903-1151, email: prem.jayaraman@infotech.monash.edu.au)
Paul Hii is a Research Student with Monash University, Melbourne, Australia 3145 (email: paul.hii@infotech.monash.edu.au)
Arkady Zaslavsky is an Associate Professor with School of Computer Sci & Software Eng, Monash University, Melbourne, Australia 3145 (email: arkady.zaslavsky@infotech.monash.edu.au)

where the devices function to deliver the best possible service to the user in a timely and most appropriate manner possible. The devices in the VPAN have enough intelligence built into them to coordinated and exchange information with its peers (others devices that provide messaging services).

The VPAN as depicted in figure 1 is a virtual environment and is a vision of the future. Devices in the VPAN are intelligent devices that have enough knowledge about the user's location and the capabilities of the other devices working within the environment. To implement a Message-On-Demand service in a UMS, the UMS need to be context-aware, terminal-aware and must be able to adapt itself based on user's personalization. Such a system working within a VPAN delivering messaging service to the user based on user demands is more realizable using a decentralized design. The devices constituting the VPAN depicted in figure 1 are decentralized which have enough intelligence to respond to user's location. This paper discusses one such decentralized architecture of a UMS that provides Messaging-On-Demand services to the user. The decentralized UMS is a pervasive [Rg00] system.

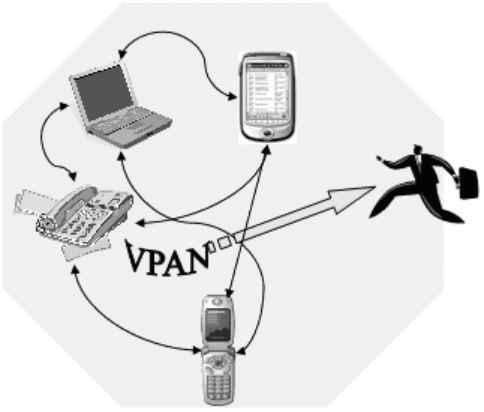


Figure 1: VPAN with user devices delivering message services to the user

Service adaptation and service personalization is one of the key factors of the modern systems, to implement a Message-On-demand service. With the evolution of pervasive computing, users have started to feel the need of service adaptation and personalization in their participating environment.

This paper presents an overview of the challenges involved in designing a decentralized UMS. A decentralized UMS tries to provide service personalization to the end user hence enabling message notification based on user demands. The system is also location and terminal aware hence, tries to provide the messaging services to the user in the most personalized manner taking into consideration the user's context, location and the end terminal which the user uses to handle the messages. The decentralized UMS also tries to overcome the challenge of delivering messages to the user in a most appropriate manner rather then just converting one message to another message format as in the traditional universal mailbox systems. The decentralized UMS functions in the VPAN in which the user is the primary objective of the system. Section 2 discusses some of the existing Unified Messaging solutions and how they differ from the system being proposed in this paper. Section 3 proposes the architecture of a decentralized UMS. It then proposes *i*-UMS, a decentralized UMS that has been prototyped as a proof of concept system. The *i*-UMS system uses wired technologies, GSM, GPRS, 802.1X wireless technologies including Bluetooth for communication between user devices for message transfer. Section 5 discusses the results obtained from the implementation.

II. UNIFIED MESSAGING SYSTEMS – AN OVERVIEW

There are a number of commercial Unified Messaging solutions that try to satisfy the goals of the unified system. Current systems depend on a gateway for message delivery which is an obvious approach for most of the UMS. These systems try to unify messages at a single end point message store by collecting messages from different message stores. The system aim at providing message delivery to the users without taking into considerations the users need. In such a system the user has minimum capability to define his reach ability from message senders. Figure 2 depicts a Unified Messaging solution provided by Eicon. Some of the other commercial UM that are available are: Nortel CallPilot [No05] [Lh03], Cisco Unity [Cu04], Global Com's Internet based UMS [Db02] and AVST Call Xpress [Av04].

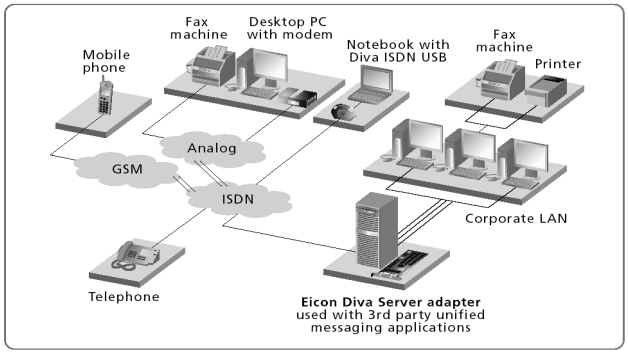


Figure 2: Eicon Diva Server [Dh03]

The basic requirement to achieve for service personalization and service adaptation is the ability to be aware of the user's environment. This primary function of being context-aware and pervasive is absent in the current UMs. The current systems also have the ability to convert messages from one form to another but they do not provide any service adaptation based on the user demands. E.g. conversion of an incoming SMS into an email and storing it in the central message store for retrieval. This example provides message conversion but does not take into consideration the user's context before converting the message into the most appropriate form. This is the same in

case of message notification. The lack of service adaptation also raises a few more questions. Some of the UM solutions use only email or SMS as the one form of message notification. But this will not satisfy the needs of the user since the user may have limited resources and hence cannot access one form of message service from another device [As99]. This approach is not always applicable in all cases especially in our VPAN where the devices work to deliver services to the user in a most appropriate manner possible. The method of using few types of message notification is not enough with the evolution of newer end-terminals and newer messaging technologies. To support this, we need to discuss the properties of few of the existing messaging technologies based on the properties show in table1 [Wj03] [Wj04]:

Time	**How long does the message needs to be stored**
Direction	**Is message Uni or Bi directional**
Audience	**The potential list of recipients**
Address	**Does system provide support for multicast, broadcast**

Table 1 : EMS properties [Wj03]

Based on the above key properties of messaging service, the following table2 [Wj03] [Wj04] gives a classification of four messaging systems and to which category they fall into.

	Email	SMS	voicemail	IM
Time	Permanent	Permanent	Temporary	Immediate
Direction	Simplex	Simplex	Simplex	Duplex
Audience	World	World	Single	Group
Address	List	List	Single	List

Table 2: Classification of four messaging systems

From the above table, it is clear that email messages fall short in the most important feature of not being duplex. It is a key factor for a pervasive messaging system to use duplex communication which can enable the user to respond immediately. Another problem is that email addresses only a list of users and not the world. Hence, using email as a method of message delivery in a pervasive environment is not viable [Wj03] [Wj04]. Also if we can consider the properties of other messaging systems, we see that each one of them fall short in one particular property. Hence we cannot use just one unique method of unification. This is where the service adaptability plays a major role. The UM must be able to adapt the message delivery based on the user's location and the end-terminal.

Another shortfall of the existing UM solutions is personalisation for the end user. The current UM solutions provides a basic level of personalisation that enables users to filter certain messages. This level of personalisation is so minimal that the system cannot personalize message delivery based on user preferences. E.g. Users can block a particular person from sending them an email irrespective of his/her location. But this is not satisfactory in terms of a UMS that delivers messages on demand. The system must be able to filter messages depending on user current environment and his environmental preferences. E.g. delivering a high priority message to the user when the user is in a meeting. The priorities are set by the users and vary depending on user's preferences. Similarly present UMs do not provide any form of service personalization. The system needs to choose the appropriate service that can be used to notify the user of a new message. As stated earlier, to implement service personalization based on user's location, the UMS must be context aware and also terminal-aware. Service adaptability and personalisation plays the key in developing a Message-On-Demand UMS. Such a system is user-centric and is pervasive. To implement such a Message-On-Demand service, the central gateway technique that the current UMs follow lacks the basic needs of being pervasive and not user-centric. Hence a system that is user-centric and pervasive can be realized by using a decentralized design rather than a gateway approach. Such a system is terminal-aware (user devices) and is context-aware, hence knowing the needs of the user providing message delivery in the most appropriate fashion possible based on user's demands. It also fulfils the goal of UMS to deliver any messages any where, any time in the most appropriate manner possible.

III. Decentralized Unified Messaging System Design and Architecture

A. System Overview

Figure 3 illustrates the overall architecture of the proposed decentralized UMS (DUMS). The DUMS is a peer to peer architecture. Each device can initiate and accept a connection request. The system receives incoming messages, notifies its peer devices about the messages, gets a response from the peer devices and replies to the sender.

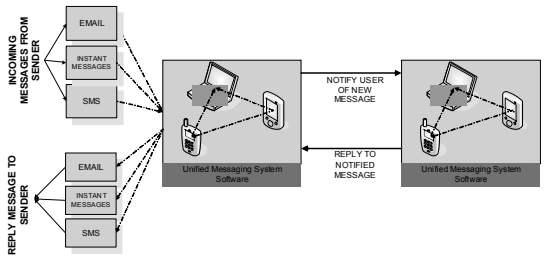


Figure 3: Overview of decentralized UMS architecture

As we can see, the system obtains messages from external sources, processes them and delivers to the user depending on user's context and end-terminal information. The devices work within the VPAN delivering messaging service on demand to the user in the most appropriate manner possible. The decentralized architecture provides the option for the user to reply to a message from any of the end-terminal irrespective of the message origination type. The DUMS takes

care of the message conversion before dispatching it.

B. Architectural Description

Figure 4 illustrates the DUMS architecture. The system, as seen, is split into a number of blocks, each of which performs a specific function. The entire system constitutes the DUMS.

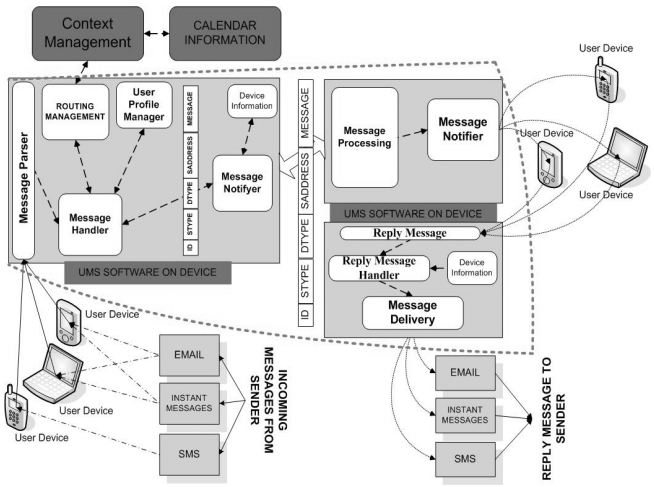


Figure 4: DUMS Architecture

The DUMS software on each device frequently checks for new messages. When a new message arrives, the incoming message handler and notifier is the first module that processes the message.

Message Parser

DUMS has the capability to convert messages from one format to another based on user's preferences and context. The message parser is a key component which can understand the various messaging systems message formats. It parses the message to obtain information like sender's address, phone number, message body, subject and other relevant information. The message parser also works on any new message notification/ response received from peer. In this case, it tries to retrieve information from the DUMS message header. The pseudo code in figure5 describes the function of the message parser.

```
if (incoming message = any message type)
then
   parse the information
   obtain sender information message body, subject
   forward this parsed information to the message
           handler
else if ( incoming message is of type DUMS
message)
then
   obtain message data from the DUMS message
   forward this information to the message handler
```

Figure 5: Message Parser

Routing Management

The routing management is the component that obtains context information, user preferences and end-terminal information to decide on the most appropriate message format and the most appropriate device to be used for message notification. The routing management comprises a routing manager that obtains context information from a context manager that is external to the DUMS. The context information provides details on user's location, his environment details and end-terminal information. Context information can be obtained by using a number of technologies most of which are still under research. Since context by itself is a major field of research, this paper does not step into the development of a robust context management system.

Message Handler

The message handler handles the inputs received from the message parser and the routing management. The message handler is the one that is responsible for the Message-On-Demand service. It involves service personalization by taking into consideration users preferences. Once message handler receives the context and end-terminal information from the routing manager, it uses the input from the profile manager to obtain user preferences. Based on user's preferences and his/her context information, the message handler decides whether or not to notify the user of the message. The data flow between the message handler and the other components is illustrated in figure6.

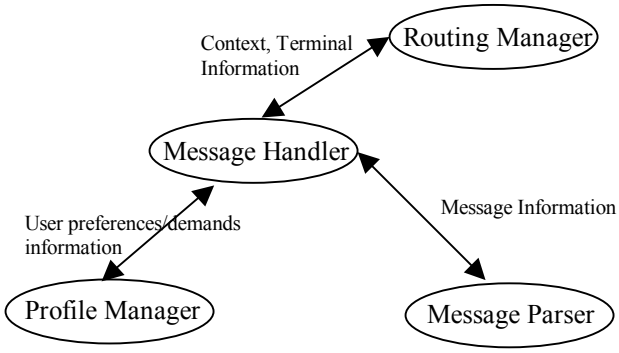


Figure 6: DFD for Message Handler, Parser and Routing Manager

Message Notifier

The message notifier handles two tasks. New message notification that needs to be passed on a peer device and notifications received from a peer device. The message notifier based on the inputs from the message handler takes into consideration, user's end-terminal device. The message notifier is responsible to establish a connection with the peer device, sending message notification to the peer device, waiting for an acknowledgement and disconnecting the connection. If the message acknowledgement is not received, the message notifier contacts the message handler to obtain information of any other device that is within the user's

environment. If the message notifier receives a message notification from the peer device, it takes into the consideration the device capability on which it is functioning and provides an appropriate notification to the user. The pseudo code illustrated in Figure7 describes the message notifier functionality.

```
Repeat
   if (message notification is PDA)    then
      notify PDA
      wait for ack.
       If ack = true
          return success
      else pass message back to message handler
   else if (message notification is smart phone) then
      notify smart phone
      wait for ack.
      If ack = true
          return success
      else pass message back to message handler
   else if (message notification is laptop) then
      notify laptop
      wait for ack.
       If ack = true
          return success
   else pass message back to message handler
while success
```

Figure 7: Message Notifier

Reply Message Handler and Delivery

The DUMS architecture provides a function that allows the user to reply to any incoming message irrespective of the device. This functionality is achieved by the Reply Message Handler and the Message Delivery components. The reply message handler is an integral component of the DUMS architecture it can handle any reply messages that the user wishes to send. The reply message handler obtains device information to check if the same device can be used to send the reply message. If it cannot use the same device, it then chooses the appropriate peer device which has the capability to send the message. For example, replying to an incoming SMS message notification as an SMS from the laptop. If the reply message handler decides to use a peer device to dispatch the reply, it uses the message delivery sub system to accomplish the task. The message delivery component uses reply message information from the reply message handler, communicates with the corresponding peer device and dispatches the user's reply message. The format of the reply message by default is the incoming message type. The pseudo code in figure8 describes the reply message handler and delivery functions.

```
Obtain and process reply message
Construct reply message format with appropriate
headers
Check reply message type
If require forwarding
Dispatch reply message to the peer device to be sent to
be dispatched to the message sender
else
dispatch reply message to message sender
```

Figure 8: Reply message handler and Delivery

C. DUMS Message Format

The system uses the DUMS message format for transferring messages between the peer devices. The DUMS message as illustrated in figure9 is made up of a set of predefined headers that encapsulates the original messages.

ID	STYPE	DTYPE	SADDRESS	MESSAGE

Figure 9: DUMS Message Format

ID: This represents the Message ID. The message ID is a unique identifier used to uniquely identify each message handled by the DUMS.

STYPE: This determines the source type of the message. The source type identifies in what format the message has actually arrived from the sender.

DTYPE: The destination type identifies in which format the message needs to be interpreted by the peer device. The DTYPE takes the same values as the STYPE. E.g., DTPYE will take value SMS if the destination device chosen to deliver the message is the users Smart Mobile Phone.

SADDRESS: This identifies the sender of the message.

IV. Implementation and Result Discussion

The DUMS architecture differentiates itself from other system by being distributed, decentralized and pervasive in nature. The system is aware of the user's devices hence, allowing the system to notify the user of new messages promptly regardless of location and time. The system is also adapts itself based on user's location and messaging preferences. The goal of the system is to send and receive any message anytime anywhere. The pervasive system requires information about user's context and environment. Context information varies depending on the environment in which the user can be in. E.g. In a meeting, at home or on the move like walking in a shopping mall or even driving a car. Since context information varies based on user location and environment, the application of DUMS is classified into a few scenarios based on the user's environment. These scenarios are:

At Home
In the office / meeting
On the Move such as driving a car

The main objective of the DUMS is to ensure prompt delivery of messages to the user in the most appropriate manner possible. It is a major challenge to implement a decentralized system. Since the design involves a number of technologies that are still under research and are at various stages of maturity, the proof of concept implementation is based on a few assumptions. Also, as discussed earlier, since context information can change depending on user's environment, the implementation to establish the DUMS architecture has been developed in specific to the Home Scenario. The system that has been developed works on the proposed DUMS architecture discussed in section3. This decentralized UMS called the intelligent-UMS (*i*-UMS) is pervasive, context-aware and provides Message-On-Demand service. The implementation of the prototype has been done using the Microsoft .Net framework. The reason behind choosing the .Net framework was, the devices, namely laptop, PDA and smart phone were all based on Windows operating system. Microsoft .Net provides rich programming api's for programming mobile devices and devices with restricted resources like the smart phone. The *i*-UMS developed was tested within the Home Scenario based on few real time situations. The following section will provide a detailed description of the Home Scenario and the various test case scenarios with along with the results obtained.

A. Home Scenario Description

Home Scenario is one of the typical cases of the usage of *i*-UMS illustrated in figure10.

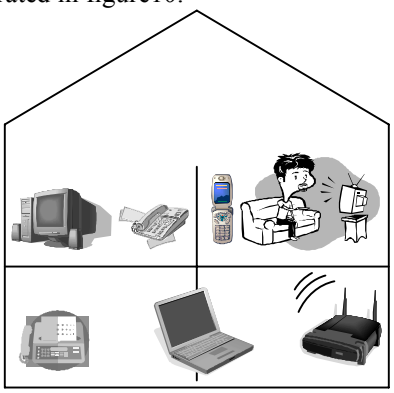


Figure 10: Home Scenario VPAN

A simple case could be: The user might be in watching television having just the mobile phone with him/her. An instant message arriving on his/her laptop from an important contact is received and processed by the *i*-UMS. The system then contacts the location manager to obtain the location information of the user and the device nearest to the user. On obtaining this information, the *i*-UMS sends a message notification to the user's mobile phone based on user's preference. The user on receiving the message has the option of replying to the message through the mobile phone. The home scenario is VPAN described previously where devices are user-centric trying to deliver messaging services to the user irrespective of time, location and message type.

B. i-UMS Implementation

The *i*-UMS prototyped was tested on the following types of message types: email, Instant Message, voicemail and SMS. The *i*-UMS makes uses of Bluetooth, GSM, IEEE 802.11 based wireless networks and wired networks for communicating between the users end-terminals. The end-terminals used for the prototype implementation were the laptop, PDA and Smart Phone. The main implementation focus was to achieve the previously discussed goal of message delivery in the most appropriate format possible irrespective of time, location and message type. To achieve this goal, the implementation will discuss mainly on the two primary functions that is used to achieve this goal. They are message retrieval and message delivery. The following sections will provide the implementation details for the laptop, smart phone and the PDA.

Message Retrieval and Parsing

The *i*-UMS on the laptop retrieves messages from the user's mailbox. The mailbox can be a POP3 server or an IMAP server. This operation is performed by using a third part tool called the PowerTCP Mail for .NET [PT04]. The *i*-UMS also retrieves IM from MSN messenger. The working of the *i*-UMS on the laptop is very simple. Whenever the messenger is set in away mode, i-UMS recognizes that the user is not near the system. Hence, it immediately monitors incoming messages. Once it receives a new message, it then retrieves the message and passes it on to the message notification module as depicted by the code in figure11. The retrieval of IM from MSN messenger is done by using the MSN messenger class module of the .NET class library. It provides rich class libraries to query and retrieve MSN messenger status and IM [DM04]. The PDA can retrieve email, IM and can notify the laptop or the smart phone depending on the context information. The functionality of the *i*-UMS on the PDA for message retrieval is almost similar to that of the laptop with a few exceptions. In home scenario, the assumption is that the laptop is connected to the network and, hence, has the ability to check for new messages frequently. Since the PDA is a low powered device, the need for the PDA to check for new messages in this scenario is not necessary. But the PDA does have the option of checking for new messages which is more applicable in case of office or on the move scenario. The message retrieval on the smart phone involves the retrieval of SMS messages. *i*-UMS on the smart phone has the added ability to retrieve any new SMS message and then parse the SMS message into a DUMS message format. Figure12 provides code snippet for monitoring and retrieving SMS messages from the smart phone.

```
Public Class Messenger
   Dim WithEvents msn As New
   MessengerAPI.Messenger
   Dim wnd1 As
   MessengerAPI.IMessengerConversationWnd
   Dim contacts As MessengerAPI.IMessengerContacts
   Dim contact As MessengerAPI.IMessengerContact
End Sub
'Event to obtain instant message
'this is a turn around method that grabs incoming
message
Private Sub msn_OnIMWindowDestroyed(ByVal
pIMWindow As Object) Handles
msn.OnIMWindowDestroyed
      Dim wnd As
MessengerAPI.IMessengerConversationWnd
       wnd = pIMWindow
       contacts = wnd.Contacts()
       Dim str As String = wnd.History
       For Each contact In contacts
          username = contact.SigninName
       Next
       If msn.MyStatus <> 2 Then
          'Construct the DUMS message
         Dim msg As New UMSData
          UMSMSG.message = msg
          If str.IndexOf(msn.MySigninName) >= 0 Then
             Exit Sub
          End If
          UMSCollection.Add(UMSMSG)
```

Figure 11: IM retrieval

```
'Retrieves new SMS message
   Private Sub sms_InboxChanged(ByVal sender As
Object,
   ByVal e As
SmartSMS.NETCF.InboxChangedEventArgs)
   Handles sms.InboxChanged
      'TextBox1.Text = " "
      Dim Msg As SmartSMS.NETCF.SMSMessage
      Dim date1 As New DateTime
      Dim date2 As New DateTime
      Dim dt() As String
      For Each Msg In e.Messages
         Dim timeString As String =
         DateTime.Now.AddMinutes(-2)
         date1 = DateTime.Parse(Msg.Time.ToString)
         date2 = DateTime.Parse(timeString)
         If DateTime.Compare(date1, date2) > 0 Then
               messageDispatcher(Msg)
         End If
      Next
   End Sub
```

Figure 12: SMS retrieval

Message Notification and Replying

Message Notification is one of the important modules of *i*-UMS. *i*-UMS does a number of processing before it actually notifies the user of the incoming message. *i*-UMS first obtains the priority of the message based on the user's profile. It then contacts the context manager to obtain the location of the user and the device that is near to the user. Once *i*-UMS figures out the appropriate device to be notified, it then checks for the availability of the devices and the technology that can be used to communicate with the device. E.g., if the device to which the message needs to be forwarded to is a PDA, then *i*-UMS checks if the PDA is in Bluetooth coverage. If it is not, then it uses wireless IEEE 802.11 to communicate with the PDA. This intelligence built into the *i*-UMS, completely makes uses of the decentralized architecture which facilitates the device to move from one location to another, but still keep in contact with the other devices. Once the notification message is sent to the user's device, the message pops up on the user's device indicating the arrival of the new message. The user can view and reply the message on the device. Once again *i*-UMS on the device uses the same intelligence to find the best possible way to forward the reply. This message dispatcher process is described in figure13.

```
Public Class MessageDispatcher
   Public Function DispatchMessage(ByVal msg As
UMSMessage) As Boolean
      Dim route As New MessageRouting
      Dim response() As String
      Dim msgSend As New SendSocket
      response = route.getContextInfo("0").Split(",")
      If response(0) = "1" Then
         System.Console.Write("Notifying the PDA")
         Return msgSend.send(msg, response(1), 22000)
      ElseIf response(0) = "2" Then
         SmsMessage(msg)
         System.Console.Write("Notifying the smart
phone")
         Return True
      End If
   End Function
End Class
```

Figure 13: Message Notification/Dispatcher

The message notification process of the *i*-UMS is same for each device. The device communicates with the context manager to obtain the location information of the user and the device, and then checks on the user's profile to see the message priority and user preferences. Once it obtains this information, *i*-UMS decides on the device which needs to be notified and sends the notification. The user has the option of replying to the message from the handheld device. E.g., the user can reply to an IM notification arriving to his/her PDA from the PDA which sends the reply back to the laptop which in turn forwards the message to the message sender. The user

has also the option of replying to the message in any format. E.g., the user can reply to an IM as SMS message.

C. Implementation Results

i-UMS was implemented based on the Home Scenario environment discussed previously. The user devices that were used for implementation were the PDA, laptop and the smart phone. The laptop is an IBM T42 laptop that runs a Windows XP professional edition. The PDA is a HP iPAQ 2200 running a Windows Mobile 2003 Operating System. The smart phone is an I-Mate smart phone 2 running a Windows Mobile 2003 smart phone Edition OS. The laptop, PDA and smart phone are all Bluetooth enabled devices. The test environment uses the context manager that is assumed to be a web service that provides *i*-UMS with the context information of the user and the user's devices. Figure14 illustrates a state transition diagram of the *i*-UMS.

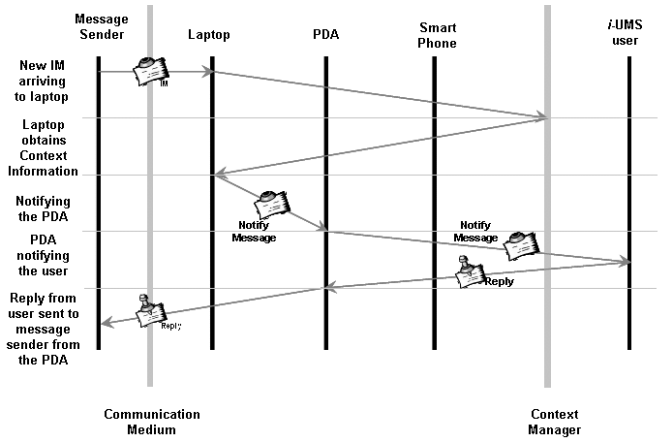


Figure 14: *i*-UMS state transition

1) Scenario 1

In this test Scenario, *i*-UMS demonstrates its message retrieval and notification functionality for an incoming instant message received from MSN Messenger. The laptop receives the new instant message, communicates with the context manager to obtain user's context information. It then obtains user preferences from the user profiler. Once the system obtains this information, it makes a decision on whether to notify the user or not. Based on the outcome of the decision, it chooses the most appropriate device to be notified based on context and user preference information. The user on receipt of the message can reply to the message from the device. The reply is then forwarded to the laptop or forwarded to the sender of the message based on the reply type. Figure15 and figure16 provides a few screenshots of the results obtained for scenario 1. In this test the destination device chosen was the PDA.

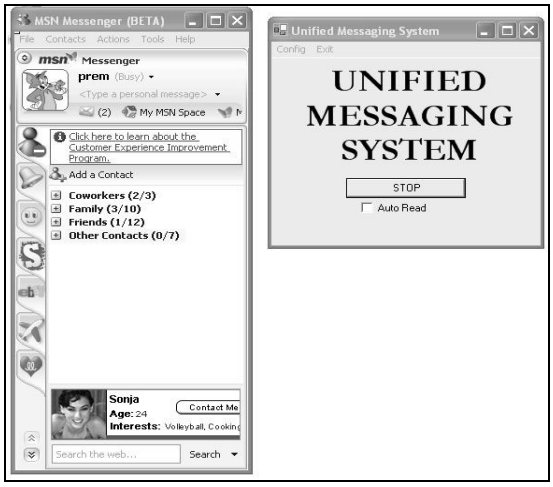


Figure 15: *i*-UMS on laptop - Instant Messenger (Busy State)

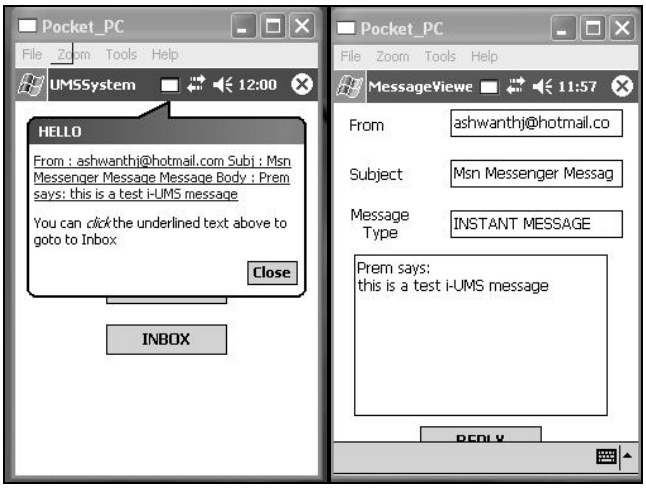


Figure 16: *i*-UMS on PDA – Receiving Notification

2) Scenario 2

Scenario 2 describes the functionality of i-UMS on the smart phone and how the system responds to any incoming SMS messages. The smart phone can retrieve SMS messages and based on user's context and user preferences, it notifies its peer device. *i*-UMS does not notify all the messages. It retrieves the messages and based on context, decides on the most appropriate device to be notified. Figure17 and figure18 provide the screen dumps of the results obtained in Scneario2. The peer device that needs to be notified in this case is the Laptop.

Figure 17: *i*-UMS on Smart Phone retrieving New SMS Message

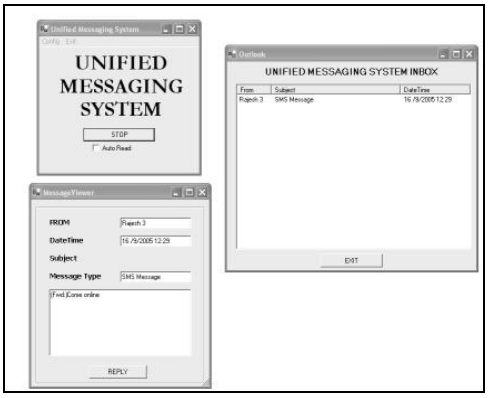


Figure 18: *i*-UMS on Laptop receiving SMS message notification

D. Evaluation Results

The evaluation of *i*-UMS is mainly based on its efficiency to handle incoming messages and message responses from the user. The *i*-UMS has been evaluated based on the following calculations.

Total Delivery Time ($\mathbf{T_d}$):

$$\mathbf{T_d} = \sum_{m=1}^{n} \mathbf{f}\,(M_{ret} + M_{prcs} + M_{delivery})$$

M_{ret} = Time to retrieve message in seconds
M_{prcs} = Time taken for message processing in seconds
$M_{delivery}$ = Time taken to deliver message to the user's device in seconds
n = Total number of messages.

Message Retrieval Time($\mathbf{M_{ret}}$):

$$\mathbf{M_{ret}} = \sum_{m=1}^{n} \mathbf{T_f}\,(M_{access} + N_c)$$

T_f = Time Function
M_{access} = Message access time. Time to access the message store in seconds
N_c = Network/ Communication Medium delays.

Message Processing Time ($\mathbf{M_{prcs}}$)

$$\mathbf{M_{prcs}} = \sum_{m=1}^{n} \mathbf{T_f}\,(M_{analysis} + T_{context} + T_{UP} + T_{decision})$$

$M_{analysis}$ = Time to parse and analyze the message in seconds
$T_{context}$ = Time to retrieve context information in seconds
T_{UP} = Time to retrieve user preferences in seconds
$T_{decision}$ = Time taken to process context and user preferences to arrive at a message delivery decision in seconds.

Message Delivery Time ($\mathbf{M_{delivery}}$)

$$\mathbf{M_{delivery}} = \sum_{m=1}^{n} \mathbf{T_f}\,(T_{MT}) + \mathbf{T_f}\,(M_{peer\ processing})$$

T_{MT} = Time for peer to respond for message transfer.
$M_{peer\ processing}$ = Time taken by the peer to obtain message notification and notify user.

The following calculation illustrates the time taken to process a reply message that is created and sent by the user. The time is measured in seconds and it considers the time taken to process the reply and dispatch the message and does not include the external time factor required to deliver the message to the message recipient.

$$\mathbf{T_{Message\ Reply}} = \sum_{m=1}^{n} \mathbf{f}\,(P_{reply} + T_{dispatch})$$

P_{reply} = Time to process the reply from the user
$T_{dispatch}$ = Time taken to dispatch the reply message(s) to the message sender. This might involve another time cost in using a peer device to deliver the message.

In Scenario 1, the system retrieves message from instant messenger and based on context and user preference information notifies the PDA of the new instant message. The total delivery time T_d calculated in this case was 11 seconds for a single message. When the system was tested for multiple messages from m=1 to 60, the average time to deliver messages increased to 13.5 sec per message. The M_{ret} is the time involved in retrieving the message which was less than a second for a single message. The message processing which involved the message analysis, retrieval of context and user information and the delivery decision took 2 seconds for a single message. Finally the message delivery which involves the time to establish connection, transfer the message and the time for the message to be processed by the peer took around 10 seconds or a single message. This was the same in case of Scenario2. Since the notification device was the laptop in Scneario2, the message processing time on the laptop was much faster when compared to the PDA. Td in this case summed up to 9.8 seconds for 1 message in queue and increased to 11.5. The results of the time to message ratio is graphically represented in figure19.

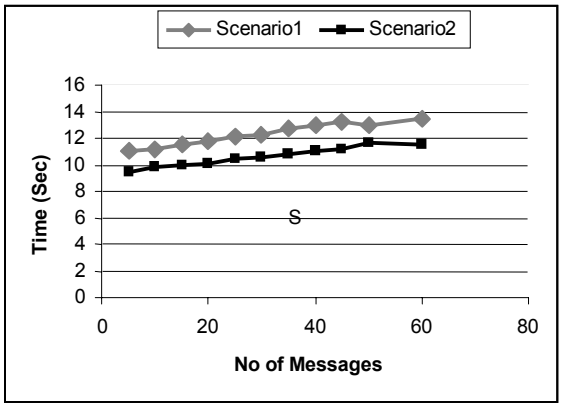


Figure 19: Performance results of i-UMS Message-Handling and Delivery Times.

V. CONCLUSION

Unified Messaging is not a completely new concept. But Message-On-Demand service is now the key one in developing pervasive messaging systems. The advent of new technologies like WLAN, Bluetooth and hardware like Personal Digital Assistant, smart phones have transformed Mark Weiser's[1] vision of pervasive computing into reality. The enormous growth of distributed computing has led to mobile computing which has allowed users to access and retrieve data irrespective of location [Sd03] [Rg00]. Such advancements in technologies have created the urge for systems that are pervasive in nature, i.e., systems that get integrated into user's environment and remain transparent.

Hence the ultimate aim of the UMS is no longer to integrate messages into one format but to deliver messaging service to users based on their demands and preferences. Service personalization and adaptation is one of the key factors to establish Message-On-Demand service. The pervasive UMS also facilitates message notification based on user's context. The system is end-terminal aware and is context aware. This kind of a service personalization can be made feasible in a decentralized design where the user's end terminals are intelligent enough to obtain user's context information. They also react to incoming messages based on the user's context. This paper has proposed DUMS, a decentralized UMS architecture that is pervasive, context-aware and user-centric. To realize the proposed DUMS architecture, the paper has presented *i*-UMS, a decentralized UMS based on DUMS architecture. The *i*-UMS has been implemented within the VPAN environment of the home scenario. Through the implementation, we have proved the feasibility of such a decentralized messaging system.

The implementation of *i*-UMS is based on a number of assumptions. These assumptions are necessary due to the primitive nature of some technologies used to realize such a system. Hence the Message-On-Demand in a decentralized UMS has still got a lot of doors open for further research. There are several extensions that we have identified to the proposed model that can render it working in the real world. Some of them are Context Awareness, Speech recognition and implementation of *i*-UMS based on DUMS architecture for the other scenarios. *i-UMS* is a next generation UMS which is pervasive, context-aware and user-centric.

REFERENCES

[As99] Arbanowski, S.; Van der Meer, S, Service personalization for unified messaging systems. Proceedings. IEEE International Symposium on Computers and Communications, ISCC'99, Read Sea, Egypt, 1999 July 6-8; Page(s): 486

[Av04] AVST, Unified Messaging in Today's Business Environment A Guide to Voice, Fax and Email Anytime, Anywhere. Vendor White Paper, 2004 September; Page(s): 20

[Cu04] Cisco Unity, Product Literature, [homepage on the Internet] Cisco Systems © 1992—2005 [Cited 20 January 2004]. Available at: http://www.cisco.com/warp/public/cc/pd-/unco/un/prodlit/

[Db02] Declan Barber, GlobalCom: A unified messaging system using synchronous and asynchronous forms. Proceedings of the inaugural conference on the Principles and Practice of programming, and Proceedings of the second workshop on Intermediate representation engineering for virtual machines, Dublin, Ireland. 2002; Page(s): 141-144. Work in progress

[Dc02] Chalmers, D.; Contextual Mediation To Support Ubiquitous Computing, In PhD Thesis, Imperial College London. 2002.

[Dh03] Daniela Horn, The Importance of an All-in-OneCommunication Adapter in a Unified Communications Solution. Eicon Networks, C 2003. Available at http://www.eicon.com/worldwide/solutions/docs/UM_WP.pdf

[DM04] DotMSN, [homepage on the Internet], Xih Solutions (c) 2003-2005. [Cited November - December 2004] Available at http://www.xihsolutions.net/dotmsn

[IA01] InfoActiv, Inc. Increasing Demand for Unified Communications. 2001, February 1: Page(s):11. Available at: http://itresearch.forbes.com/detail/RES/992971989_481.html

[Lh03] Larry Hettick, Inside Unified Messaging: A comprehensive shopper's guide. Nortel Networks. Copyright© 2003. Available at http://www.nortelnetworks.com/products/01/callpilot/collateral/nn106800-021104.pdf

[Mj04] Michael James, Liz Rice. An Introduction to Unified Messaging. Internet Applications Group, Data Connection Limited, 2004 August.

[No05] NORTEL, Nortel CallPilot Unified Messaging, Product Brief, 2005 Available at:http://www.nortel.com/products/01/callpilot/collateral/nn101801-041105.pdf

[PT04] Power TCP Mail for .NET, [homepage on the Internet] Dart Communications, © 2004. [Cited November - December 2004]. Available at: http://www.dart.com/dotnet/mail.asp

[Rg00] Grimm, R.; Anderson, T.; Bershad, B.; and Wetherall, D.; A System Architecture for Pervasive Computing. Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system, Kolding, Denmark. 2000; Page(s): 177 – 182

[Sd03] Saha D, Mukherjee A. Pervasive computing: a paradigm for the 21st century. IEEE Publication, 2003 March.

[Vd99] Van der Meer .S, Arbanowski. St, Magendanz T, An approach for a 4th generation messaging system. Proceedings. The Fourth International Symposium on Autonomous Decentralised Systems, Tokyo, 1999 March.

[Vs00] Van der Meer .S, Arbanowski. St, Steglich. St, Flexible control of media gateways for service adaption. Proceedings of IEEE Intelligent Network Workshop, 2000 May

[Wj03] Wams, J.M.; van Steen, M, Pervasive Messaging. Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003). 2003 23-26 March; Page(s): 499 - 504

[Wj04] Wams, J.-M.S.; van Steen, M. Unifying user-to-user messaging systems. IEEE Internet Computing. 2004 March-April; Page(s): 76 – 82

[1] Source from Saha D, Mukerjee, Pervasive computing: a paradigm for the 21st century

Haggle: a Networking Architecture Designed Around Mobile Users

(Invited Paper)

James Scott*, Pan Hui†, Jon Crowcroft†, Christophe Diot‡
*Intel Research Cambridge
james.w.scott@intel.com
†Cambridge University
firstname.lastname@cl.cam.ac.uk
‡Thomson Research
christophe.diot@thomson.net

***Abstract*— Current mobile computing applications are infrastructure-centric, due to the IP-based API that these applications are written around. This causes many frustrations for end users, whose needs might be easily met with local connectivity resources but whose applications do not support this (e.g. emailing someone sitting next to you when there is no wireless access point). We identify the general scenario faced by the user of Pocket Switched Networking (PSN), and discuss why the IP-based status quo does not cope well in this environment. We present a set of architectural principles for PSN, and the high-level design of Haggle, our asynchronous, data-centric network architecture which addresses this environment by "raising" the API so that applications can provide the network with application-layer data units (ADUs) with high-level metadata concerning ADU identification, security and delivery to user-named endpoints.**

I. Introduction

End user experiences of mobile, many-device computing are often marked by frustration and inconvenience. Users are forced to be highly aware of their connectivity environment, with many applications only working when networking infrastructure is available. One ubiquitous example is that of two people with laptops sitting next to each other, who cannot email a file they wish to share because infrastructure is either unavailable, not working properly, or too costly. While there are other ways to send the file, this requires training and further understanding of the network situation – most often, people simply fall back on the use of USB key flash drives. The billion US dollar market for these in 2005 [1] is a testament to the failure of the mobile networking research community to provide a network architecture that supports truly mobile applications.

In this paper, we make the following contributions aimed towards beginning to address that failure. We first provide a formulation of the networking environment faced by mobile users, a scenario which we term Pocket Switched Networking (PSN) (Sect. II), and describe how the status quo of TCP/IP is unable to cope with PSN (Sect. III). We present a set of architectural principles which we believe will enable a network architecture to perform well under PSN conditions (Sect. IV). We then describe Haggle, our clean-slate design for a new network architecture following these principles (Sect. V). Finally, we present related work (Sect. VI) and conclusions (Sect. VII).

II. Pocket Switched Networking

In designing a new network architecture, it is first important to define the scenario in which that architecture will be used. IP, for example, was designed against a backdrop of a multitude of existing networks, and with the primary needs being resilient end-to-end communications in the presence of node failures, as befits its originator, the US Department of Defense [2].

Pocket Switched Networking (PSN) is the term we use to describe the situation faced by today's mobile information user. Such users have one or more devices, some/all of which may be with them at any time, and they move between locations as part of a normal schedule. In so moving, the users can spend some (or much) of their time in "islands of connectivity", i.e. places where they have access to infrastructure such as 802.11 access points (APs) which they can use to communicate with other nodes via the Internet. They also occasionally move within wireless range of other devices (either stationary or carried by other users) and are able to exchange data directly with those devices.

Thus, in PSN, there are three methods by which data can be transferred, namely neighborhood connectivity to other local devices, infrastructure connectivity to the global Internet, and user mobility which can physically carry data from place to place. For the former two methods, the connectivity is subject to a number of characteristics, including those of bandwidth, latency, congestion, synchronicity (e.g. email or SMS are asynchronous, while ad-hoc 802.11 is synchronous), the duration of the transfer opportunity (i.e. the time till the device moves out of range), and also monetary cost (usually only for infrastructure). For the latter method of user mobility, users acting as "data mules" can transfer significant amounts of data, and while users' movements cannot in general be controlled, they can be measured [3], and patterns in those movements can be exploited.

In addition to the issue of network connectivity, we must also consider the usage model for PSN. While different applications have different network demands, we can distinguish particular broad classes which are known to be useful: (a) *known-sender* where one node needs to transfer data to a user-defined destination. The destination may be another user (who may own many nodes), all users in a certain place, users with a certain role (e.g. "police"), etc. The key point is that, often, the destination is not a single node but is instead a set of nodes with some relationship, e.g. the set of nodes belonging to a message recipient. (b) *known-recipient* in which a device requires data of some sort, e.g. the current news. The source for this data can be any node which is reachable using any of the three connectivity types, including via infrastructure (e.g. a news webpage), neighbours (e.g. a recent cache of a news webpage) or mobility (e.g. the arrival of a mobile node carrying suitable data). In both classes described above, the endpoints of a network operation are no longer described by network-layer addresses, but are instead a set of desirable properties. As a result, general network operations no longer have single source and destination nodes.

Finally, in PSN situations, resource management is a key issue. Mobile devices have limited resources in terms of storage, network bandwidth, processing power, memory, and battery. The latter is perhaps the most important, since the others can potentially be reclaimed without the user's assistance, while charging the battery requires the user to perform the physical act of plugging it in, and restricts the device's mobility while charging. Other resources are also precious, particularly in the face of demands imposed by the usage scenarios above, where devices may need to use storage and network bandwidth to help forward messages for other devices. However, there is also much cause for optimism — storage capacities are increasing exponentially, wireless networking has the useful property of spatial reuse, and processing power on mobile devices is growing with Moore's Law. For power, many devices are plugged in more often than not, e.g. notebook computers, and low power electronics allows current mobile phones to last for many days on a single charge.

From the discussion above, we extract three motivations for a networking architecture in the PSN environment, in order of importance:

- Allow applications to take advantage of all types of data transfer (neighborhood, infrastructure, mobility) without having to specifically code for each circumstance
- Allowing networking endpoints to be specified by user-level naming schemes rather than node-specific network addresses, thus each network operation can potentially involve many endpoints.
- Allowing limited resources to be used efficiently by mobile devices, taking into account user-level priorities for tasks.

III. Problems with status quo

Current applications perform very badly in the PSN environment, since they are typically designed around some

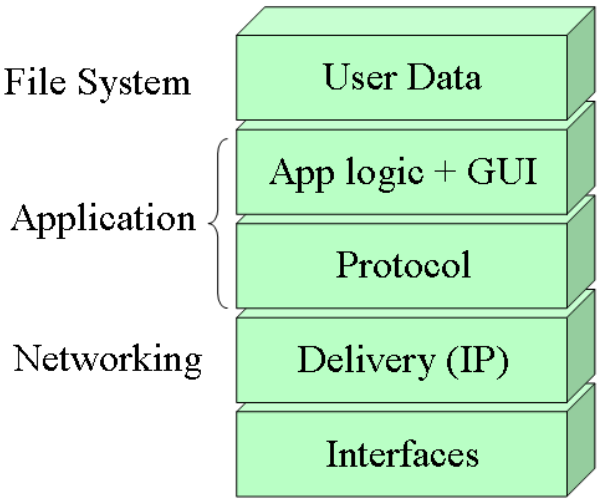


Fig. 1. Current networking architecture for mobile applications

form of infrastructure which is not always available. While some applications can cope with infrastructure blackout, e.g. with a "disconnected" or "offline" mode, most do not. Direct, neighbourhood connectivity is used by very few widely used applications, and human mobility is deliberately used by almost none. Thus, when infrastructure is not present, users are presented with huge inconveniences since the applications which are familiar to them stop working, and are forced to take on the task of understanding these situations so that they can be productive despite this application failure. For instance, users may require many alternative applications in order to do a single task depending on the situation, e.g. a file can be exchanged by email, by putting it on a website for download, by using an instant messaging client, by direct Bluetooth or infrared transfer. More likely, the user will simply invest in a USB key — and manually bypass the huge inconvenience of the status quo.

The root cause of this is the fact that applications are provided with a networking interface that only understands streams of data directed at anonymous numeric endpoints (namely TCP/IP). As illustrated in Figure 1, this forces developers to implement protocols for naming, addressing and data formatting internally in the applications themselves, e.g. SMTP, IMAP and HTTP. While at the GUI level, applications have general user-level tasks such as "send this file to James," once a particular network protocol such as SMTP is imposed on that task, it becomes the a more specific task, e.g. "send this file to the server pointed at by the MX record in the DNS record of the domain name part of james.w.scott@intel.com". The latter task is specific to a particular kind of connectivity scenario, in this case infrastructure-based. It is therefore impossible to execute even if James's device is in the neighbourhood at that time — i.e. even if the user-level task could be satisfied.

Another problem with the current networking API is that it is synchronous. Applications cannot indicate a network task to be performed and then exit, since finished applications have all their TCP/IP sockets closed. For example, an email application with pending outgoing email in the outbox will not be able to use a passing AP to send this email if the application is not running when the AP is passed. Therefore, an application in the PSN environment has to be constantly on and monitoring the connectivity status of the device. This increases the complexity of a disconnection-aware application, since it must be able to wait through periods of bad connectivity and detect and perform networking actions when a suitable endpoint is again visible. It also increases the load on mobile device resources, since many applications would have to be present in the background at all times.

A third problem is that persistent user data is kept by applications in a file system which, in the current node architecture, is unconnected from the networking system (again illustrated in Figure 1). This means that all "sharing" of data between nodes must often be conducted by applications themselves[1]. The biggest example of this is the device synchronisation problem — when a user has multiple devices, they must explicitly run an application on each which pulls their data out of the file system and shares it with their other device(s). Such synchronisation is often a source of much inconvenience for users, since the sync tools must understand the different ways that each user application uses the file system to store data and metadata, and often has to translate it so that different applications can be sync'd with the same data. Another example is in distributed web caching — the exact web page that a user wants may be in the cache of a neighbouring node, but since web browsers do not explicitly support the transfer, there is no way to get this off the neighbour's file system and into the network to be shared with the user.

The final problem identified is that applications have no easy way to prioritise the use of a mobile devices' limited resources. These resources include persistent storage, network bandwidth, and battery energy. Currently, an application such as a web browser must estimate by itself how much of the storage can be used for non-critical history caching, or how much network bandwidth should be used for pre-fetching of web pages. This decision is often passed on to the user, who might have to adjust settings manually, at the application level (e.g. "how much disk to use as cache"), at the hardware level (e.g. turning on or off wireless network interfaces depending on the battery level), or by only running certain apps when they do not want to prioritise network bandwidth for other tasks (e.g. network-hungry file-sharing apps). These controls are coarse at best, and require expert understanding in order to properly exercise them. The result is that resources are often used inefficiently.

[1]Networked file systems can be used for data sharing, but these rely heavily on good connectivity, often to a particular server, and as such are not generally usable in PSN

IV. A New Set of Mobile Networking Principles

We now present a set of interrelated principles which we believe are fundamental implications of the situation faced by users' devices in the PSN environment, and can provide solutions to the problems with the status quo. These guide the design of the Haggle architecture presented below, but are also applicable to other architectures for any networking scenario with similar characteristics. Note that we do not claim that the individual principles below are novel, some (such as message switching) are very well-known. We do believe that we are the first to assemble this particular *set* of principles.

A. Forward using application layer information

Applications should not be forced to specify endpoints using addresses, such as IP addresses, that are meaningless at the user level. Instead, endpoints should be specified using higher-layer information, e.g. the URL of a website. By performing forwarding with such information, Haggle can satisfy the application's needs using any form of connectivity — e.g. going to that URL directly if there is infrastructure, or obtaining it directly from a neighbour who has a cached copy, or using a node known to be passing an AP soon to physically carry the request and propagate the answer back, etc. In other words, we need to move from *node-centric* networking to *data-centric* networking.

Taking this example a step further, website URLs are often found using search engines, whereas the user's request is actually for information matching a particular set of keywords. Haggle can use such keywords directly, e.g. a request for "current world news" can be satisfied with cached copies in the environment of any news website (perhaps with a user-specified order of preference, or a whitelist/blacklist).

Similarly, using an email address for forwarding restricts a messaging application to using email protocols and infrastructure, while using a phone number restricts the application to forwarding using SMS. By allowing Haggle to use the person's name (the higher-level, more meaningful identifier), it can use any protocol for which it has a mapping between the high-level name and a protocol-meaningful address.

B. Asynchronous operation

Asynchronicity is important in three ways in Haggle. First, applications should be able to indicate networking actions asynchronously from the actions taking place. This is in contrast to the current model where applications must be "on" throughout the transmission (as described in Section III), and thereby reduces the complexity of a PSN-friendly application. Second, this also means that the decision of precisely which next-hop node(s) to forward data to can be left as late as possible; in other words, the forwarding algorithm can use "late binding" when assigning a low-layer next hop address, allowing it to best utilise up-to-date local context information about which next-hop nodes allow the data to make the most progress toward a destination. Third, asynchronicity is key to the store-and-forward nature of Haggle, which allows it to cope with non-contemporaneous connectivity between

endpoints in a way that end-to-end protocols such as TCP cannot.

C. Empower intermediate nodes

In PSN, intermediate nodes (i.e. nodes on the transmission path that are not specified as destinations in the data being transferred) may also be valid destinations for data. For example, if a mobile device acts as a forwarding node for a webpage, that device may wish to keep a cache of the webpage in case its own user later wishes to view it, or it comes into contact with another device which requires that information. This is in contrast to infrastructure-based networking where intermediate nodes do not usually reconstruct application-layer data to decide whether it is locally useful, and the data is simply transmitted end-to-end.

This is effectively ad-hoc multicasting, where the multicast group can be joined at any time by any device which can see (or get to) a copy of the data. It has significant advantages over demand-driven data transmission — since a demand for a data item at a particular location (with no infrastructure) cannot be transmitted easily to a node which is moving towards that location and has a chance to pick that data up. However, if that node opportunistically stores the data, perhaps using policies or learning algorithms to determine whether it is likely to be "popular", then the data can arrive unbidden at a location where it is useful.

D. Message switching

All of the above three principles imply that message switching is more suitable than packet switching for Haggle. This is not to say that the underlying networks might not use packet switching, but that full application-level messages should be exchanged by neighbouring Haggle nodes when possible. Message switching means that application-layer forwarding information does not have to be duplicated across many packets, it facilitates asynchronous operation by the networking subsystem, and it means that intermediate nodes are provided with the whole message so they can act as destinations as well as forwarding points for any given message.

E. All user data kept network-visible

Asynchronicity implies that user data in transit needs to be kept in a node's Haggle framework. However, we take this principle further: In PSN, *all* user data should be made visible to Haggle at all times. In addition, data must be marked with metadata about its user-level properties, such as access authorisation, creation/modification/expiry times, etc. We have two main reasons for this.

First, a significant fraction of user data is inherently shared, i.e. the user's task involves making it available to other users according to some access control profile. For example, CVS file stores are shared by many users via an infrastructure-based communication model. By making all user data visible to Haggle, such data can be transmitted to other authorised users without relying on infrastructure, making CVS-like applications capable of running under general network conditions. Note that we do not tackle the general data merging/versioning problems that CVS does, but that we simply provide a means for the communications part to be abstracted.

Second, users often have more than one device. Therefore, even a user's most private data should be network-visible, if only for transmission to other devices that they own (or devices that they trust, e.g. an Internet-based backup service). Currently, data synchronisation between multiple user devices is a very thorny problem both for the developers of such tools, and for users who have to manually associate devices that they want synchronised. We can alleviate some part of this problem by making sure all user data is visible to Haggle and marked with information on who is allowed to access it.

By making all user data visible to the network, we decouple the data from particular nodes and allow it to flow to the set of nodes with a valid interest in it. With Haggle, we aim to achieve this in the face of flexible connectivity environment inherent in PSN.

F. Build request-response into the network

In IP, there is no notion of a "request" for data at a layer lower than the application. However, many user-level tasks (and therefore applications) make use of request-response semantics, e.g. web browsing or file sharing. In PSN situations, we often need to locate data of interest using dynamic and local connectivity rather than at a static infrastructure-based location. However, if requests and responding to requests were not part of the network, then we have situations as with the status quo where a webpage that I want may be on a computer next to me, but there is no way for my computer to ask for that webpage without relying on a particular peer-to-peer filesharing application being active.

To take another example, a mobile node might be at a location where it has no infrastructure connectivity, but it may wish to facilitate incoming data from other nodes, e.g. so it can receive email, or retrieve updates for the local web cache. By sending a request, it can cause other nearby nodes, which may for example have infrastructure connectivity, to act on its behalf, and eventually have the resulting messages propagated back towards it. This can lead to significant resource savings – if the requests include information on the current connectivity situation (e.g. sender's location, nearby nodes, path the request took), then the responses can be directed more quickly and/or with a lower level of message replication (since successful delivery of each replica is more likely when using up-to-date network state information).

G. Exploit all data transfer methods

The aim of Haggle is to take advantage of all the communication opportunities offered by the PSN environment, including local connectivity with neighbouring nodes, and global connectivity using infrastructure. Human mobility patterns can be exploited by using forwarding algorithms which target nodes known to have mobility patterns which are likely to be useful, e.g. because they have seen a destination node

recently [4], or because they share the same mobility pattern as a destination [5].

Between neighbouring nodes, there are potentially many interfaces that can be used, e.g. two neighbour nodes might have a Bluetooth, 802.11 ad-hoc mode, and infrared as potential connection opportunities. Haggle nodes must maintain a mapping of interfaces to nodes, since it is wasteful to for two nodes to exchange data multiple times using different interfaces. Each connectivity method may have different properties in terms of bandwidth, latency, power consumption, etc, as well as having time-dependent channel characteristics such as congestion, so the choice of the correct connection method may be dynamic.

Infrastructure connectivity is not uniform either, with a particular infrastructure method having associated costs (which may be per-byte, per-time, or more complex schemes with varying rates, e.g. mobile phone contracts with a certain number of "free text messages" per month), as well as bandwidth, connection setup latency, per-message latency, etc. Some infrastructure methods are synchronous, e.g. when using direct TCP/IP between two Haggle nodes connected to the Internet. Some are asynchronous, e.g. the use of SMS text messages which are held until the recipient's phone is on and has cell tower coverage. Haggle has to cope with both of these types.

H. Take advantage of brief connection opportunities

In the PSN scenario, connection opportunities can be fleeting, e.g. when walking or driving past another mobile user or an AP. It is therefore important for Haggle to be able to take full advantage of time-limited connection opportunities, by prioritising potentially exchanged data so that the most urgent data is sent first, and by using underlying protocols which make efficient use of bandwidth during short connection opportunities [6]. This also implies that neighbourhood discovery (neighbours meaning both mobile devices and APs in this case) is a key part of Haggle, since transfer opportunities must be detected in a timely fashion.

I. Empowered and informed resource management

Many of the principles above refer to resource management in some sense. Resource management is key to the success of Haggle since many of the proposals above have the potential to use up unlimited amounts of resources, e.g. data that is currently being held and forwarded for other nodes requires storage space, network bandwidth to send and receive, processing power to make and effect transfer decisions, and battery power to do all of the above.

This resource consumption might well be viewed a potential disadvantage of Haggle; for example, if a Haggle user's device were to run out of battery because it spends it all on forwarding others' data, that user will quickly disable Haggle. In fact, Haggle offers a unique opportunity to build resource management in to mobile devices in a scalable way, with minimal overhead for applications. Mobile devices often have plentiful resources. With battery life, many devices such as laptops have a "portable" rather than mobile usage model, and are plugged in when they are on. Storage capacity is growing at an exponential rate, with gigabytes already the reality, but typically users have to manually decide to copy data objects onto devices and personally manage the use of this large resource. Wireless networks have the great advantage of spatial reuse, but often only the space around APs is used, and away from APs there is much unused bandwidth being wasted, despite mobile devices moving through those spaces.

1) Storage: Storage resources are currently often used on a "first come first served" basis, and are only filled when applications specifically request it. This leads many devices to have most of the disk empty, so that they are "overprovisioned", and for devices which do run out of disk, the user often needs to manually find and remove low-importance data such as web caches. Haggle, since it keeps all user data, has the potential to manage storage space better, since some data is of clearly higher priority than others. For example, Haggle could flag each data item as "manual deletion only" (e.g. a document being edited), "delete if absolutely necessary" (e.g. a local cache of the user's old photo collection) or "deletion okay" (e.g. the web cache), and with priority levels, so that the data Haggle is holding in transit for a stranger is less valued than data held for someone who regularly communicates with the user of the device, i.e. a friend.

2) Networking: Networking resources must be managed for two reasons. Firstly, as discussed above, a particular connection opportunity may be of limited duration, and it is therefore important to prioritise the data sent using that scarce bandwidth so it is used to obtain the greatest benefit from the user's perspective. The second reason is that a given Haggle node may have an almost unlimited set of networking tasks on its "to-do list", due to transfers in progress, as well as the need to use the network medium for neighbour discovery. To blindly execute the networking tasks in parallel or in FIFO order, as often done by network stacks now, will lead to low user-level goodput under high load (much of which may be speculative and replicated transmissions). Therefore, networking tasks should be carried out in an order determined by user-level priorities.

3) Battery: Battery resources are perhaps the most important to manage properly on a mobile device, as once spent they cannot be recovered without user intervention. Mismanagement can cause many problems for users, e.g. the inability to rendezvous with friends/family if your phone "dies." Users are therefore very protective of their battery life, and if Haggle (and PSN technologies in general) are perceived to be battery-thirsty this might prove to be a key roadblock to deployment. It is less obvious is that, in some situations, users have *plenty* of battery resource. For example, while in a normal weekday routine, many people can easily charge my phone at night since there is a charger by their bedside. Since many phones normally require charging only once every few days, there is plenty of energy available if the users do not mind charging them every night in return for better application performance. Similarly, many laptops move from being plugged-in at one

place to plugged-in at another, and are only on battery power for short periods of time.

For battery in particular, it is important to determine the "scarcity" of the resource — i.e. an estimate of how long it will be until the next convenient charging opportunity for the user. This can be achieved using *context-awareness* — by observing the patterns that the user exhibits at various times of day, the device's location, etc, a device can apply machine learning techniques to arrive at a prediction of how scarce the battery resource is. Thus, even if Joe User's battery is full, if Joe leaves his home city and heads to the airport, Joe's devices could infer that there may be no charging opportunity for some time, and therefore be conservative with battery consumption. Conversely, if Joe User's battery is only at 10%, but Joe will be home in 10 minutes and habitually plugs their device in on arrival at home, then perhaps there is plenty of battery to use for even low-priority application tasks.

Because Haggle has the ability to centrally manage all networking and storage consumption of a mobile node, it is also the correct place for battery consumption to be managed and where control over its consumption can be exerted — e.g. a particular connection opportunity might be deliberately unused because Haggle determines that the utility gained does not outweigh the battery cost.

J. Use and integrate with existing application infrastructure where possible

Haggle is not intended as an academic exercise in network architecture design, it is intended to be practical and useful. We must therefore pay close attention to existing deployments of applications and infrastructure for these applications, and integrate and reuse these. Haggle can gain three key advantages from doing so. First, Haggle is more easily incrementally deployed to users if they can interact with other users who do not yet have Haggle, via backwards compatibility. Second, users may wish to continue using the same, familiar application interfaces that they already make use of. Third, there is a vast infrastructure already deployed that will not change overnight, which Haggle must make use of in order to be competitive with the status quo.

V. The Haggle Architecture

Haggle is the name given to our new network architecture, which applies the principles outlined above to overcome the problems with the status quo and effectively operate in the PSN environment. Haggle is an unlayered architecture which internally comprises four modules; delivery, user data, protocols and resource management, as shown in Figure 2. As compared to the current network architecture in Figure 1, we immediately notice a number of high-level differences:

- User data is not isolated from the network, allowing it to be shared with other suitable nodes without an application being involved in each transfer
- The application does not include network protocol functionality, making it easy for it to be agnostic as to the delivery method, and making the application code simpler.
- Haggle performs delivery using user-level names, allowing it to make use of all suitable protocols and network interfaces for delivery of a given data item.
- Haggle includes a resource management component which mediates between the other three components using user-specified priorities to ensure efficient resource use.

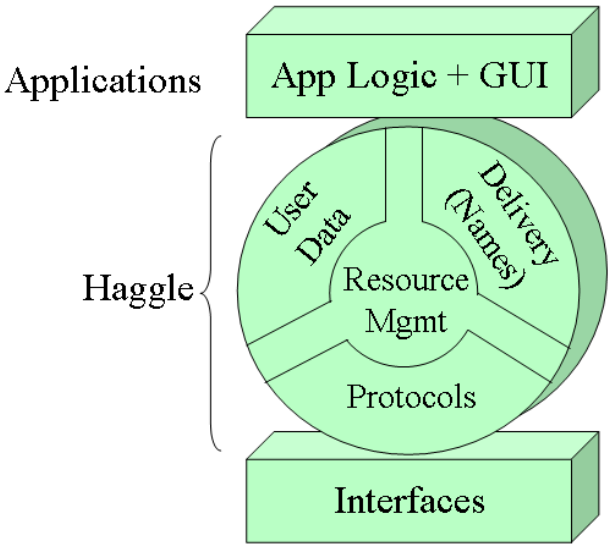


Fig. 2. Overview of Haggle architecture

We now discuss the functionality of the four modules in more detail.

A. User Data

Application Data Units (ADUs) are Haggle's format for user data, and as the name suggests they are an encapsulation for a data item meaningful to an application such as a photo, a music file, a webpage, a message, etc. In the spirit of Application Layer Framing [7], ADUs are not only the unit of data for applications, but are also the unit used for transfer between Haggle nodes, i.e. they are the messages in message switching. An ADU is comprised of many attributes, where an attribute is a type-value pair. The type is always a text string, the value may also be a text string, but may also be a binary stream, e.g. for the mp3 file comprising the main content of an ADU representing a song. The intent is for rich metadata about each object to be exposed as attributes — such data could be duplicated in the binary stream section if this is useful for the application.

An ADU attribute can be used to store such information as:

- Descriptive information for user data, e.g. keywords for a picture.
- Document management information, e.g. the creation-date, creation-user, modification-date, etc.

- Security and permissions information, e.g. the list of local users with access permission, whether the item must be encrypted on leaving the node, and information to the encryption method and key

Novel forwarding algorithms are a key research area for Haggle, and the ADU structure provides flexible support for forwarding algorithms. This is achieved by storing forwarding state in ADUs as well as application data described above. Unlike with packet headers in traditional protocols, ADUs are a flexible data format that can easily be modified to add or remove fields.

Thus, ADUs can also contain information about forwarding state such as:

- a list of destination names and addressing hints for those names (there can be more than one destination, each with more than one way to get there).
- details of the source node.
- a list of forwarding hints, informing intermediate nodes of dynamic information (e.g. X-was-seen-at-time-T-by-Y)
- a list of nodes which it has passed through
- timeout/flooding avoidance parameters — e.g. max duplication count, max hop count, deadline
- a priority level specified by the source node
- security information, e.g. an authentication signature or encryption details for the claimed ADUs
- any other data relevant for forwarding tasks — since the ADU format allows a variable number of attributes, a new forwarding algorithm can easily send other data.

1) ADU Example: The example ADU below represents a message including a photo, some text, and forwarding state for the message.

Filename	DSC10027.jpg
Mime-Type	image/jpeg
Creation-date	1/1/2006 17:32
Created-by	James Scott
Security-group	Public
Keywords	Athens, Greece, seashore, sunset
Data	[binary jpeg data]
Mime-type	text/plain
Date	1/1/2006 17:40
Data	”Wish you were here!”
Date	1/1/2006 17:40
From	“James Scott”; james.w.scott@intel.com
To	“Jon Crowcroft”; +447123456789; jon.crowcroft@cl.cam.ac.uk
To	“Pan Hui”; pan.hui@cl.cam.ac.uk; BT 0F:CC:3E:C9:87:21
Priority	5
TTL-hops	100
TTL-deadline	2/1/2006 17:40

Note that this ADU comprises three sections, similar to emails with different message parts. Thus, the photo could be easily split off from the rest of the message, e.g. by an intermediate node whose user had specified an interest in photographs of sunsets, while the whole message was delivered. In implementation, the photo itself would not be duplicated on a node’s disk, but instead pointers to the same data file would be used for efficiency.

B. Protocols and Naming

A key feature of Haggle is the use of user-level naming schemes for data transfer decisions. This immediately raises the question of how these high-level names get translated into lower-level addresses that the physical network interfaces can use for transmission. In other words, what is the equivalent of ARP for Haggle?

In order to answer this question, we must examine what an “address” is for Haggle. We define an address as any name for which there is a protocol available in the Protocols module (see Figure 2) which is capable of sending the ADU to that address. Different connectivity situations and different applications require different address types, for example, the use of local connectivity to share ADUs with a neighbour might use a Bluetooth or 802.11 MAC address as a Haggle address, while when using infrastructure, an email address can be used as a Haggle address (if an email protocol is supported). Note that different nodes might regard different types of name as “addressable” because of their different capabilities — to a node with geographic information and forwarding capabilities, a GPS coordinate is an addressable name, while to another node that name needs to be mapped onto another name type before forwarding can occur. Mechanisms for mapping between names are therefore very important in Haggle.

There are many methods by which names can be mapped to other names. An ADU can contain mappings between names, e.g. the ADU example given above, which specified mappings between a non-addressable personal name, and email, telephone and Bluetooth MAC addresses which various underlying protocols could use. Names can also be dynamic, e.g. the current IP address of an infrastructure-connected mobile node, which changes from time to time, or the current location of a node (if a geographic routing protocol is available). Both static and dynamic name mappings may be found in the ADU itself, or may be found other ADUs acting as name lookup tables on the local node. Such naming ADUs might be created by applications (e.g. contact details for a particular person) or by forwarding algorithms (e.g. using neighbourhood device discovery). The use of a standard ADU format for naming has the benefit of allowing different nodes, applications, and forwarding module implementations to parse naming ADUs from each other. The flexible nature of ADUs also means that different naming schemes can be constructed easily, enabling the use of Haggle for new applications and protocols in a way that highly-specified lookup tables (e.g. DNS MX records) do not.

C. Neighbours and Forwarding

As previously mentioned, Haggle must perform neighbour discovery in order to take advantage of connection opportunities. The result of neighbour discovery is that some set of addresses are marked as "nearby". One example might be Bluetooth inquiry, which would result in a set of Bluetooth MAC addresses being marked as "nearby", while another example would be that when an accessable AP is seen, all Internet domain names would be considered "nearby". Forwarding algorithms in Haggle estimate the "benefit" of performing a transfer of a given ADU (or set of linked ADUs) to a given nearby node. While some transfers are obviously beneficial, e.g. the transfer of an ADU to an email address which it lists as a destination, other transfers are less obvious, e.g. the transfer of the same ADU to a node which is not the final recipient but might be willing to help in the transfer process, or the transfer of a ADU requesting content to a particular neighbour who may or may not help provide the content requested.

D. Resource Management

All use of resources in Haggle is controlled by the resource management module. This operates by performing a cost/benefit analysis on "tasks" that other modules specify. The forwarding module, for example, specifies a number of potential transfers as "tasks" with associated benefit estimates, and also gives enough information so that the "cost" of those tasks can be estimated in terms of resources consumed, including the use of network bandwidth, battery power, monetary cost, etc.

The resource management module compares the cost with the benefits to decide what action to take next. In addition, the resource management module can use context-based information to estimate the *scarcity* of resources, e.g. the network bandwidth available, the expected time until battery charging can take place, etc. This can be incorporated into the cost-benefit decision by raising or lowering the value placed on certain resources dynamically.

By performing resource management centrally, we allow applications to cooperate in sharing resources rather than competing, since applications can specify priority levels for various actions and allow low-priority actions to avoid using scarce resources. We can also provide the user with the ability to specify global preferences for the device, e.g. regulating the consumption of expensive bandwidth to an acceptable level.

E. Interacting with applications

The Haggle architecture provides a new abstraction layer for mobile applications, at a much higher level than the "socket" abstraction that is currently used. The key properties of the Haggle architecture from the application's point of view are:

- Haggle supersedes the file system on mobile devices, providing a persistent storage abstraction for Application Data Units (ADUs) which allows applications to specify a rich set of metadata governing how that data is used.
- Haggle provides applications the ability to specify networking tasks based on the contents of ADUs, e.g. asking Haggle to retrieve ADUs that are photos of a particular place and time, or webpages with certain keywords.
- Haggle abstracts the networking facilities of the mobile device so that the application does not need to implement particular transfer protocols which are infrastructure-centric, and can instead transparently make use of neighbourhood, infrastructure, and mobility-based data transfer opportunities.
- Haggle provides a way of naming and addressing entities which are not single network nodes (as with IP) but are high-level concepts such as people, places, services, or information.
- Haggle allows applications to specify priorities for tasks, so that the limited resources of mobile devices can be spent for maximal user benefit, and spare resource can be used for secondary tasks (e.g. web prefetching) while not jeopardising the primary tasks (e.g. urgent messaging).

VI. Related work

In this section we discuss related work by ourselves and by others.

As with many pieces of research, the proposed architecture above creates as many questions as it answers. There are many challenges faced in PSN [8]. These include the problems of designing forwarding algorithms for the PSN environment, of creating suitable naming schemes and mapping those names onto deliverable addresses, of security and privacy protection, and of usability when there are no end-to-end guarantees.

Our approach towards these challenges is practical rather than theoretical, using implementation, deployment, and measurement. The architecture design above is being implemented for mobile platforms such as mobile phones and PDAs, and will be tested with real applications and real users. This will allow us to hone the architecture to address real-life situations, and, in collaboration with others, to address the various challenges detailed above.

Human mobility for data transfer has been explored by a number of different research groups, including under the names of "data mules" [9] and "message ferries" [10]. In Haggle, our approach has been to perform measurements of human mobility patterns [3], [11]. These have found that human mobility has significantly different characteristics to those assumed in simulations which have previously been used to evaluate neighbourhood forwarding algorithms, e.g. in mobile ad-hoc networks which have relatively dense networks. Such measurement traces can be used to help design and evaluate forwarding algorithms for the PSN environment.

Delay-Tolerant Networks (DTN) [12], [13] focuses on protocols addressing scenarios where TCP/IP networking is not feasible, with two such scenarios being when there are large time delays (e.g. in interplanetary networks), or when there is no contemporaneous end-to-end link, e.g. when using a

"message ferry" to physically carry data to remote locations. Haggle shares some of its principles with DTN, such as the use of message switching and opportunity-oriented networking, but is additionally exploring ideas such as the mapping of user-level names onto many parallel delivery methods, the exposure of all user data to the network, the use of request and response as network primitives, and the key role of resource management.

The data-centric networking aspect of Haggle is similar in nature to a number of efforts, including FreeNet [14] and Distributed Hash Tables (DHTs) such as Chord [15] in which a hash of an object is used to locate an object, and peer-to-peer networks such as eMule which allow text searches over metadata such as the file name to find objects. In Haggle, we aim to perform data-centric networking in the PSN environment, which does not have the relatively stable connectivity assumed by the previous work.

VII. Conclusions and Future Work

Haggle is a network architecture designed from the ground up around the needs of mobile users as characterised by the Pocket Switched Networking environment. We have described the motivation for a new architecture, and the principles behind Haggle's design. We plan to build an open source, cross-platform implementation of Haggle for mobile devices, and trial this implementation with both new and existing applications (via translating proxies when necessary). We also plan to use Haggle to develop and evaluate solutions to various challenges in PSN, including forwarding algorithms, security policies, usability aids, and resource management policies.

Acknowledgments

The authors would like to thank Augustin Chaintreau at Thomson and Richard Gass at Intel for insightful comments in discussions on this work. This work has been supported by the European Union under the Haggle integrated project FP6-IST-027918, and by the ACCA coordination action FP6-IST-6475.

References

[1] Gartner, Inc., "Market trends: Usb flash drives," http://www.gartner.com/DisplayDocument?doc_cd=130007.

[2] D. D. Clark, "The design philosophy of the DARPA internet protocols," in *Proceedings of ACM SIGCOMM (Computer Communications Review Vol 18, No 4)*, 1988.

[3] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket Switched Networks and human mobility in conference environments," in *Proceedings of the SIGCOMM 2005 Workshop on Delay-Tolerant Networking (W-DTN05)*. ACM.

[4] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *Proceedings of The First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, August 2004.

[5] J. Leguay, T. Friedman, and V. Conan, "Evaluating mobility pattern space routing for DTNs," in *Proc. INFOCOM*, 2006.

[6] R. Gass, J. Scott, and C. Diot, "Measurements of in-motion 802.11 networking," in *IEEE WMCSA 2006 (to appear)*, 2006.

[7] D. Clark and D. Tennenhouse, "Architectural considerations for a new generation of protocols," in *ACM SIGCOMM*, 2000.

[8] P. Hui, A. Chaintreau, R. Gass, J. Scott, J. Crowcroft, and C. Diot, "Pocket Switched Networking: Challenges, feasibility and implementation issues," in *Proceedings of the Workshop on Autonomic Communications*, ser. LNCS, vol. 3457. Springer-Verlag, 2005.

[9] S. J. Rahul C Shah, Sumit Roy and W. Brunette, "Data mules: Modeling a three-tier architecture for sparse sensor network," in *IEEE Workshop on Sensor Network Protocols and Applications (SNPA)*, May 2003.

[10] M. A. Wenrui Zhao and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *ACM Mobihoc*, May 2004.

[11] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *Proceedings of IEEE INFOCOM*, 2006.

[12] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of ACM SIGCOMM*, 2003.

[13] Delay Tolerant Networking Research Group, "http://www.dtnrg.org/."

[14] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," *Lecture Notes in Computer Science*, vol. 2009, 2001. [Online]. Available: citeseer.ist.psu.edu/clarke00freenet.html

[15] R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *ACM SIGCOMM 2001*, San Diego, CA, September 2001.

A Layer-2 Architecture for Interconnecting Multi-hop Hybrid Ad Hoc Networks to the Internet

E. Ancillotti
Dept. of Information Engineering
University of Pisa
Via Diotisalvi 2 - 56122 Pisa, Italy
Email: emilio.ancillotti@iet.unipi.it

R. Bruno, M. Conti, E. Gregori, A. Pinizzotto
IIT Institute
National Research Council (CNR)
Via G. Moruzzi, 1 - 56124 PISA, Italy
Email: {r.bruno,m.conti,e.gregori,a.pinizzotto}@iit.cnr.it

***Abstract*— Recently, the research on mobile ad hoc networks is departing from the view of stand-alone networks, to focus on hybrid self-organized network environments interconnected to the Internet. This type of networks is built on a mix of fixed and mobile nodes using both wired and multi-hop wireless technologies, and may be easily integrated into classical wired/wireless networking infrastructures. In this paper we design a lightweight and efficient architecture to build such a multi-hop hybrid ad hoc network, which will be used as a flexible and low-cost extension of traditional wired LANs. Our proposed architecture provides transparent global Internet connectivity and self-configuration to mobile nodes, without requiring configuration changes in the pre-existing wired LAN. Differently from most of the implemented solutions, which are based on complex IP-based mechanisms, such as Mobile IP, IP-in-IP encapsulation and IP tunneling, our proposed system operates below the IP level, and employs only layer-2 mechanisms. We have prototyped the core functionalities of our architecture, and we present several experimental results to verify the network performance constraints, and how different OLSR parameter settings impact on them.**

I. Introduction

A mobile ad hoc network (MANET) is a collection of mobile nodes connected together over a wireless medium, which self-organize into an autonomous multi-hop wireless network. Traditionally, MANETs have been considered as *stand-alone* networks, i.e., self-organized groups of nodes that operate in isolation in an area where deploying a networking infrastructure is not feasible due to practical or cost constraints (e.g., disaster recovery, battlefield environments). However, it is now recognized that the commercial penetration of the ad hoc networking technologies requires the support of an easy access to the Internet and its services. In addition, the recent advances in mobile and ubiquitous computing, and inexpensive, portable devices are further extending the application fields of ad hoc networking. As a consequence, nowadays, multi-hop ad hoc networks do not appear as isolate self-configured networks, but rather emerge as a flexible and low-cost extension of wired infrastructure networks, coexisting with them. Indeed, a new class of networks is emerging from this view, in which a mix of fixed and mobile nodes interconnected via heterogeneous (wireless and wired) links forms a multi-hop hybrid ad hoc network integrated into classical wired/wireless infrastructure-based networks [1].

In this paper, we propose and evaluate a practical architecture to build multi-hop hybrid ad hoc networks used to extend the coverage of traditional wired LANs, providing mobility support for mobile/portables devices in the local area environment. More precisely, we envisage a hybrid network environment in which wired and multi-hop wireless technologies transparently coexist and interoperate. In this network, separated group of nodes without a direct access to the networking infrastructure form *ad hoc "islands"*, establishing multi-hop wireless links. Special nodes, hereafter indicated as *gateways*, having both wired and wireless interfaces, are used to build a wired backbone interconnecting separated ad hoc components. To ensure routing between these ad hoc parts, a proactive ad hoc routing protocol is implemented on both gateways' interfaces. In addition, the gateways use their wired interfaces also to communicate with static hosts belonging to a wired LAN. The network resulting from the integration of the hybrid ad hoc network with the wired LAN is an *extended* LAN, in which static and mobile hosts transparently communicate using traditional wired technologies or ad hoc networking technologies.

In this work we specifically address several architectural issues that arise to offer IP basic services, such as routing and Internet connectivity, in the extended LAN. First, we propose a dynamic protocol for the self-configuration of the ad hoc nodes, which relies on DHCP servers located in the wired part of the network, and it does not require that the ad hoc node to be configured has a direct access to the DHCP server. In addition, we design innovative solutions, which exploit only *layer-2* mechanisms as the ARP protocol, to logically extend the wired LAN to the ad hoc nodes in a way that is transparent for the wired nodes. More precisely, in our architecture the extended LAN appears to the external world, i.e., the Internet network, as a single IP subnet. In this way, the hosts located in the Internet can communicate with ad hoc nodes inside the extended LAN as they do with traditional wired networks. Previous solutions to connect ad hoc networks to the Internet have proposed to use access gateways that implement Network Address Translator (NAT) [2] or a Mobile IP Foreign Agent (MIP-FA) [3]. However, such approaches are based on complex IP-based mechanisms originally defined for the wired Internet, like IP-in-IP encapsulation and IP

tunneling, which may introduce significant overheads and limitations, as discussed in depth in the following sections. On the other hand, the architecture we propose in this paper is a lightweight and efficient solution that avoids these overheads operating below the IP level. By positioning our architecture at the layer 2 (data link layer), we may avoid undesired and complex interactions with the IP protocol and provide global Internet connectivity and node self-configuration in a very straightforward way.

In the past, other architectures have been proposed to provide ad hoc support below IP. For example, in [4] label switching was employed to put routing logic inside the wireless network card. More recently, the LUNAR [5] ad hoc routing framework and the Mesh Connectivity Layer (MCL) [6] have been proposed. These solutions locate the ad hoc support between the layer 2 (data link layer) and layer 3 (network layer). This "layer 2.5" is based on *virtual* interfaces that allow abstracting the ad hoc protocols from both the specific hardware components and network protocols. However, this interconnection layer requires its own naming and addressing functionalities distinct from the layer-2 addresses of the underlying physical devices. This may significantly increase the packet header overheads. On the contrary, our proposed architecture is totally located inside layer 2, reducing implementation complexity and ensuring minimal additional overheads.

We have prototyped the main components of our architecture in a general and realistic test-bed, in which we have carried out various performance measurements. The experimental results show the performance constraints with mobility and Internet access, and indicate that an appropriate tuning of the routing protocol parameter may significantly improve the network performance. The rest of this paper is organized as follows. Section II introduces our network model. In Section III, we discuss available solutions for Internet connectivity and node self-configuration in MANETs. Section IV outlines the relevant protocols used in our architecture. In Section V, we describes the layer-2 architecture we propose to build hybrid ad hoc networks interconnected to the Internet. Section VI shows experimental results on the performance constraints of our solution. Finally, Section VII draws concluding remarks and discusses future work.

II. Network Model

Figure 1 illustrates the reference network model we assume in our architecture. We consider a full-IP network in which all the traffic is transported in IP packets. In this network, mobile/portable nodes far away from the fixed networking infrastructure establish multi-hop wireless links to communicate (e.g., using IEEE 802.11 technology). As shown in the figure, gateways, i.e., nodes with two interfaces - both wired and wireless - are used to connect the ad hoc components to a wired LAN (e.g., an Ethernet-based LAN). In our architecture, it is allowed the multi-homing, i.e., the presence of multiple gateways within the same ad hoc component. Consequently, specific mechanisms are required to support the handoff between gateways without TCP-connection breaks. In general, between pairs of gateways in radio visibility of each other, two direct links can be established, both wired and wireless. However, in our model we assume that the gateways always use the wired link to communicate. The motivations behind this requirement will be clearly discussed in Section V. However, this is a quite reasonable assumption, since wired links have higher bandwidth than wireless links, and the routing protocol should assign them a lower link cost.

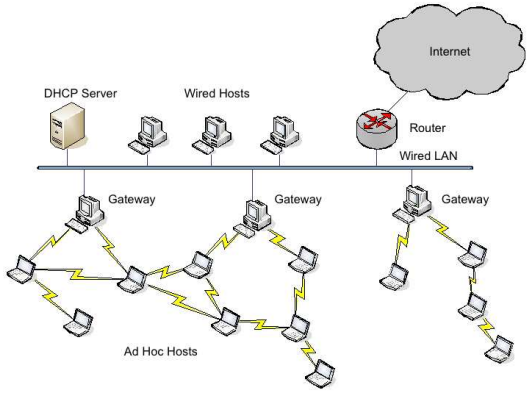


Fig. 1. Reference Network Model.

The wired LAN is interconnected to the external Internet through a default router R. In addition, one or more DHCP servers are located in the wired LAN to allocate network addresses to hosts. In the following sections, we will explain how these DHCP servers could be used to assign IP configuration parameters also to the ad hoc nodes. For the purpose of simplicity, we assume that all the IP addresses are allocated from the same IP address block $\mathrm{IP_S/L}$. According to standard notation, $\mathrm{IP_S}$ indicates the network prefix, and L is the network mask length, expressed in bits (e.g., $\mathrm{IP_S/L} = X.Y.96.0/22$). Assuming that the extended LAN adopts a unique network address implies that the extended LAN appears to the external world, i.e., the Internet network, as a single IP subnet.

Standard IP routing is used to connect the extended LAN to the Internet. However, a specific ad hoc routing protocol is needed to allow multi-hop communications among the ad hoc nodes. In this work we decided to use a proactive routing protocol as the ad hoc routing algorithm (such as the Optimized Link State Routing (OLSR) protocol [7] or the Topology Dissemination Based on Reverse-Path Forwarding (TBRF) routing protocol [8]). The motivation behind this design choice is that proactive routing protocols usually support gateways, allowing these nodes to use special routing messages to set up default routes in the ad hoc network. Indeed, default routes are an efficient mechanism to forward traffic that does not have an IP destination locally known to the ad hoc network. In addition, proactive routing protocols, adopting classical link state approaches, build the complete network-topology knowledge in each ad hoc node. This topology information could significantly simplify the operations needed to acquire

Internet connectivity. In this work, the reference ad hoc routing algorithm is OLSR, but our architecture is general and it is equally applicable to other proactive routing protocols.

III. Related Work

The implemented solutions to provide Internet connectivity in MANETs are mainly based on two different mechanisms.

One approach is to set up a Mobile IP Foreign Agent (MIP-FA) in the gateway and to run Mobile IP [3] in the MANET. In this way, the ad hoc node may register the foreign agent care-of-address with its Home Agent (HA). Whenever an ad hoc node MN wants to contact an external host X, it uses its home address (i.e., a static IP address belonging to its home network) as source address. As a consequence, the return traffic is routed to the home network through standard IP routing. The HA intercepts the traffic, encapsulating it using the care-of-address, and it tunnels the encapsulated packets to the FA. The FA removes the outer IP header and delivers the original packets to the visiting host MN. Different versions of this approach have been proposed and implemented for proactive [9] and reactive [10] ad hoc networks. A drawback of these solutions is that they require significant changes in the Mobile IP implementation since the FA and the mobile node cannot be considered on the same link. Moreover, the mobile node has to be pre-configured with a globally routable IP address as its home address, limiting both the ability of forming totally self-configuring and truly spontaneous networks, and the applicability of these schemes.

An alternative solution to interconnect MANETs to the Internet is to implement a Network Address Translation (NAT) [2] on the gateway. In this way, the gateway may translate the source IP address of outgoing packets from the ad hoc nodes with an address of the NAT gateway, which is routable on the external network. The return traffic is managed similarly, with the destination IP address (i.e., the NAT-gateway address) replaced with the IP address of the ad hoc node. NAT-based solutions have been designed for both proactive [11] and reactive [12] ad hoc networks. NAT-based mechanisms appear as easier solutions than MIP-FA-based schemes to provide Internet access to MANETs. However, a problem that arises with NAT-based solutions is multi-homing, i.e., the support of multiple gateways in the same MANET. Indeed, to avoid session breakages it is necessary to ensure that all the packets from the same session are routed over a specific gateway. A proposed solution to this issue is to explicitly tunnel all the outgoing traffic from the same communication session destined to the external network to one of the available gateways, instead of using default routes. A limitation of this strategy is the additional overhead introduced by the IP-in-IP encapsulation. Moreover, the ad hoc nodes should be provided with the additional capability of explicitly discovering the available gateways. This would eventually require extensions to the ad hoc routing protocols.

Both the two classes of solutions discusses above implicitly assume that either there is a dynamic host configuration protocol designed to configure the nodes such as to properly working in the MANET, or the ad hoc nodes are configured *a priori*. Indeed, a node in an IP-based network requires a unique IP-based address, a common netmask and, eventually, a default gateway. In traditional networks, hosts rely on centralized servers like DHCP [13] for configuration, but this cannot be easily extended to MANETs because of their distributed and dynamic nature. However, various protocols have been proposed recently in literature for the purpose of address self-configuration in MANETs. In general, with protocols using stateless approaches nodes arbitrarily select their own address, and a Duplicate Address Detection (DAD) procedure is executed to verify its uniqueness and resolve conflicts. On the other hand, protocols based of stateful approaches execute distributed algorithms to establish a consensus among all the nodes in the network on the new IP address, before assigning it. The protocols proposed in [14] and [15] are examples of the latter and former approach, respectively, while [16] presents a general overview of the several solutions currently available. Generally, all these protocols assume reliable flooding in order to synchronize nodes' operations and resolve inconsistencies in the MANET, but this is difficult to be guaranteed in ad hoc networks. Another main limitation of these solutions is that they are designed to work in *stand-alone* MANET, while no protocols have been devised to take fully advantage of the access to external networks. In addition, the problems of selecting a unique node address, routing the packets and accessing the Internet are still separately addressed, while a unified strategy may be beneficial, reducing complexities and overheads.

IV. Protocol Descriptions

This section gives a short description of the protocols, which our architecture is based on.

A. OLSR

The OLSR protocol [7], being a link-state proactive routing protocol, periodically floods the network with route information, so that each node can locally build a routing table containing the complete information of routes to all the nodes in the ad hoc network running on their interfaces the OLSR protocol. The OLSR routing algorithm employs an efficient dissemination of the network topology information by selecting special nodes, the multipoint relays (MPRs), to forward broadcast messages during the flooding process. The link state reports, which are generated periodically by MPRs, are called Topology Control (*TC*) messages. MPRs grant that TC messages will reach all 2-hop neighbors of a node. In order to allow the injection of external routing information into the ad hoc network, the OLSR protocol defines the Host and Network Association (*HNA*) message. The *HNA* message binds a set of network prefixes to the IP address of the node attached to the external networks, i.e., the gateway node. In this way, each ad hoc node is informed about the network address and netmask of the network that is reachable through each gateway. In other words, the OLSR protocol exploits the mechanism of *default routes* to advertise Internet connectivity.

For instance, a gateway that advertises the $0.0.0.0/0$ default route, will receive all the packets destined to IP addresses without a known route on the local ad hoc network.

B. ARP Protocol

IP-based applications address a destination host using its IP address. On the other hand, on a physical network individual hosts are known only by their physical address, i.e., MAC address. The ARP protocol [17] is then used to translate, inside a physical network, an IP address into the related MAC address. More precisely, the ARP protocol broadcasts the *ARP_Request* message to all hosts attached to the same physical network. This packet contains the IP address the sender is interested in communicating with. The target host, recognizing that the IP address in the packet matches its own, returns its MAC address to the requester using an unicast *ARP_Reply* message. To avoid continuous requests, the hosts keep a cache of ARP responses.

In addition to these basic functionalities, the ARP protocol has been enhanced with more advanced features. For instance in [18] it has been proposed the *Proxy-ARP* mechanism, which allows constructing local subnets. Basically, the Proxy ARP technique allows one host to answer the ARP requests intended for another host. This mechanism is particularly useful when a router connects two different physical networks, say *NetA* and *NetB*, belonging to the same IP subnet. By enabling the Proxy ARP on the router's interface attached to *NetB*, any host A in *NetA* sending an ARP request for a host B in *NetB*, will receive as response the router's MAC address. In this way, when host A sends IP packets for host B, they arrive to the router, which will forward such packets to host B.

V. Proposed Architecture

Our design goal in the definition of the rules and operations of the proposed architecture is to provide transparent communications between static nodes (using traditional wired technologies) and mobile nodes (using ad hoc networking technologies), employing mechanisms that run below the IP layer. As discussed in the introduction, in this work we address two relevant issues: node self-configuration and global Internet connectivity.

A. Ad Hoc Node Self-configuration

The main obstacle to use a DHCP server for self-configuration of ad hoc nodes is that the DHCP server may be not reachable to the new node, due to mobility or channel impairments. In addition, the ad hoc nodes may need multi-hop communications to reach the DHCP server, but a unique address is necessary to execute ad hoc routing algorithms capable of establishing such communications. To solve these problems, we assume that the DHCP servers are located only in the wired part of the network, while in the ad hoc part of the network we implement dynamic *DHCP Relay* agents. These are special relay nodes passing DHCP messages between DHCP clients and DCHP servers that are on different networks. As illustrated in Figure 2, when a new mobile host i not yet configured attempts

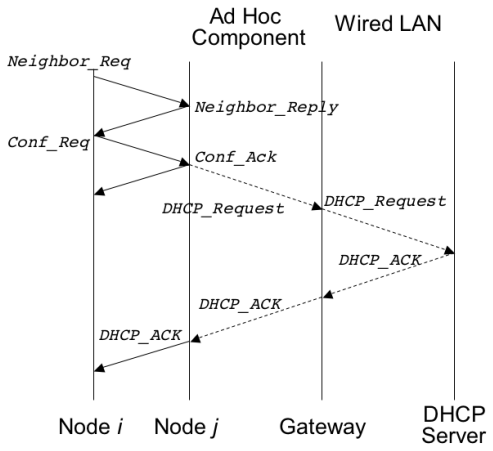


Fig. 2. Message exchanges during the ad hoc node self-configuration.

joining the ad hoc part of the extended LAN, it broadcasts a special message, the *Neighbor_Req* message. At least one neighbor that is already configured, i.e., it has joined the ad hoc network, will respond with a *Neighbor_Reply* message. Node i selects one of the responders j as intermediary in the process of address resolution. Then, node i sends a *Conf_Req* message to the chosen node j that replies with a *Conf_Ack* message to inform node i that it will execute on its behalf the process of acquiring the needed IP configuration parameters (i.e., node j acts as a *proxy* for the node i). In fact, on receiving the *Conf_Req* message from node i, node j activates its internal DCHP Relay agent, which issues an unicast *DHCP_Request* to one of the available DHCP servers. The DHCP server receiving the request, will answer to the DHCP Relay with a *DHCP_Ack*, containing the IP configuration parameters. The configuration process is concluded when the DHCP Relay forwards the *DHCP_Ack* message to the initial node i that is now configured and can join the network. After joining the network, node i may also turn itself into a DHCP Relay for the DHCP server from which it received the IP configuration parameters, letting other nodes to subsequently joining the ad hoc component. Finally, it is worth noting that it is not needed any initialization procedure for the ad hoc network, because the gateways are directly connected to the wired LAN and can broadcast a *DHCP_Discover* message to locate available servers. In this way, the first mobile node to enter the ad hoc network may find at least one gateway capable of initiating the illustrated configuration process.

Our proposed node self-configuration mechanism is somehow similar to the one described in [15]. In that paper, a preliminary message handshake was used to discover a reachable MANET node that could act as initiator of the configuration process. On the contrary, in our solution the initiator node exploits the resources of the external wired network to which the ad hoc component is connected, to perform the IP address resolution.

B. Global Internet Connectivity

Our design goal is to support Intranet connectivity (i.e., communications with nodes inside the same IP subnet) and Internet connectivity (i.e., communications with nodes of external IP networks) for the mobile nodes, without any configuration change in the pre-existing wired LAN. The assumption that we take as starting point in our proposal is that the mobile nodes are configured with an IP address belonging to the same IP subnet of the wired LAN. This is achieved using the mechanism described in Section V-A.

In the following we will separately explain how the proposed architecture ensures connectivity for outgoing and incoming traffic.

1) Connectivity for Outgoing Traffic.: As outlined in Section IV-A, the OLSR protocol builds the routing tables with entries that specify the IP address of the next-hop neighbor to contact to send a packet destined to either another host or subnetwork. More precisely, to send a packet to a destination IP address, the mobile host searches for the longest IP prefix in the routing table matching the destination IP address. The matching routing table entry provides the next hop to send the packet. Since the gateways advertise $0.0.0.0/0$ as default route, all packets destined for IP addresses without a specific route on the ad hoc network, will be routed on a shortest-hop basis to the nearest gateway and forwarded to the Internet. However, using $0.0.0.0/0$ as default route for outgoing packets, introduces an inconsistency when a mobile host sends IP packets to a wired host inside the LAN. To explain this problem

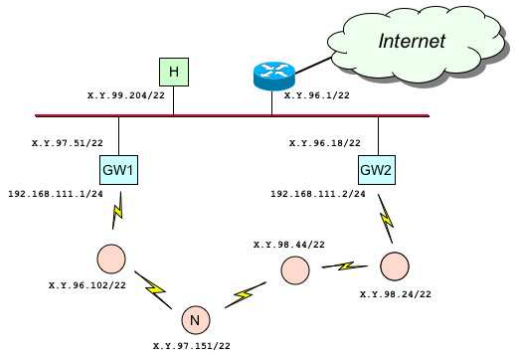


Fig. 3. Illustrative network configuration.

let us consider the simple network configuration depicted in Figure 3. For illustrative purposes we assume that the IP subnet of the extended LAN is $\mathrm{IP_S/L} = X.Y.96.0/22$[1]. If the mobile node N ($\mathrm{IP_N} = X.Y.97.151/22$) wants to deliver packets to the wired node H ($\mathrm{IP_H} = X.Y.99.204/22$), the routing table lookup on node N will indicate that the node H is connected to the same physical network of node N's wireless interface. This will result in a failed ARP request for the $\mathrm{IP_H}$ address. To resolve this inconsistency, we will exploit the properties of the IP longest-matching rules. More precisely, we split the original IP subnet into two consecutive smaller subnets $\mathrm{IP_{SL}/(L+1)}$ and $\mathrm{IP_{SU}/(L+1)}$, such as to have that the union of these two sets is equal to $\mathrm{IP_S/L}$. In the considered case $\mathrm{IP_{SL}/(L+1)} = X.Y.96.0/23$ and $\mathrm{IP_{SU}/(L+1)} = X.Y.98.0/23$[2]. Then, we configure all the gateways in such a way that they announce, through the HNA messages, also the connectivity to these two subnetworks. In this way, each mobile host will have, for any host on the local wired LAN, a routing table entry with a more specific network/mask than the one related to its wireless interface. To better clarify this point, let us consider the node N's routing table as shown in Table I. The entries 8, 9, and 11 are the ones induced by the HNA messages arrived from GW1. The entry 10 is automatically set up by the operating system when the wireless interface is configured with the IP parameters. However, when searching the routing table for matching the $\mathrm{IP_H}$ address, node N will found the routing entry 9 more specific than entry 10. Consequently, the longest-match criterion applied to the routing table lookup, will result in node N correctly forwarding traffic to gateway GW1 (i.e., the nearest one) to reach node H.

TABLE I
NODE N'S ROUTING TABLE.

Entry	*destination*	*next hop*	*metric*	*interface*
1	$X.Y.97.51/32$	$X.Y.96.102$	2	$eth0$
2	$X.Y.96.102/32$	0.0.0.0	1	$eth0$
3	$X.Y.98.44/32$	0.0.0.0	1	$eth0$
4	$X.Y.98.24/32$	$X.Y.98.44$	2	$eth0$
5	$X.Y.96.18/32$	$X.Y.96.102$	3	$eth0$
6	192.168.111.1/24	$X.Y.96.102$	2	$eth0$
7	192.168.111.2/24	$X.Y.96.102$	3	$eth0$
8	$X.Y.96.0/23$	$X.Y.96.102$	2	$eth0$
9	$X.Y.98.0/23$	$X.Y.96.102$	2	$eth0$
10	$X.Y.96.0/22$	0.0.0.0	0	$eth0$
11	0.0.0.0/0	$X.Y.96.102$	2	$eth0$
12	127.0.0.0/8	127.0.0.1	0	$l0$

The mechanism described above resolves any eventual IP inconsistency that could occur in the mobile hosts, but it may cause problems for the gateways. In fact, being part of the ad hoc component, the gateways will receive HNA messages sent by other gateways, setting up the additional routing entries advertised in these messages. However, when a gateway wants to send packets to a wired host on the local wired LAN (e.g., node H), the routing table lookup will choose one of these two entries, instead of the entry related to its wired interface (i.e., $X.Y.96.0/22$). The effect is that the IP packet will loop among the GW nodes until the TTL expires, without reaching the correct destination H. To resolve this problem, we statically add in each gateway two further routing entries in addition to the one related to the default router $X.Y.96.1$. These two additional entries have the same network/mask as the two announced in the HNA messages, but with lower

[1]On the gateways' wireless interfaces we set up private IP addresses to save address space. In this way, the gateways are globally reachable using the IP address on their wired interfaces.

[2]It is straightforward to observe that this operation is always feasible, at least for $L < 32$.

metric. Again, to better clarify the routing operations, let us consider the illustrative example shown in Figure 3. In Table II we have reported the GW1's routing table. In this example, $eth0$ is the GW1's wireless interface and $eth1$ is the GW1's wired interface. When gateway GW1 wants to send packets to node H, it will found two routing table entries matching the same number of bits of node H's IP address. These are entry 9 (derived from HNA messages received from GW2) and entry 11 (statically configured on the gateway). However, entry 11 has a lower metric than entry 9 (i.e., metric 0 against metric 1). As a consequence, the packets destined to host H can be correctly forwarded to the host H on the local wired LAN through the GW1's wired interface.

TABLE II
GW1'S ROUTING TABLE.

Entry	*destination*	*next hop*	*metric*	*interface*
1	$X.Y.96.102/32$	0.0.0.0	1	$eth0$
2	$X.Y.97.151/32$	$X.Y.96.102$	2	$eth0$
3	$X.Y.98.44/32$	$X.Y.96.18$	3	$eth1$
4	$X.Y.98.24/32$	$X.Y.96.18$	2	$eth1$
5	$X.Y.96.18/32$	0.0.0.0	1	$eth1$
6	192.168.111.2/24	$X.Y.96.18$	1	$eth1$
7	192.168.111.0/24	0.0.0.0	0	$eth0$
8	$X.Y.96.0/23$	$X.Y.96.18$	1	$eth1$
9	$X.Y.98.0/23$	$X.Y.96.18$	1	$eth1$
10	$X.Y.96.0/23$	0.0.0.0	0	$eth1$
11	$X.Y.98.0/23$	0.0.0.0	0	$eth1$
12	$X.Y.96.0/22$	0.0.0.0	0	$eth1$
13	0.0.0.0/0	$X.Y.96.1$	0	$eth1$
14	0.0.0.0/0	$X.Y.96.18$	1	$eth1$
15	127.0.0.0/8	127.0.0.1	0	$l0$

2) Connectivity for Incoming Traffic.: A mechanism is required to ensure that the return traffic coming from hosts on the local wired LAN or from the Internet (through the default LAN router, as shown in Figure 1), gets correctly routed to the mobile hosts. Our basic idea is to introduce specific Proxy ARP functionalities into each gateway, in such a way that the gateways can hide the ad-hoc node identity on the wired physical network, which the gateways are connected to. Thus, all mobile nodes located in the ad hoc component will appear to wired hosts as being one IP-hop away. Internally to the ad hoc component, the ad hoc routing protocol will transparently provide the multi-hop connectivity and the mobility support. This is somehow similar to what is implemented in the LUNAR framework [5], in which the entire ad hoc network appears as a single virtual Ethernet interface.

In our proposed solution, a Proxy ARP server runs on the wired interfaces of each gateway. The Proxy ARP server periodically checks the gateway's routing table and ARP table, such as to publish the MAC address of the gateway's wired interface for each IP address having an entry in the routing table with a netmask $255.255.255.255$, and the next hop on the gateway's wireless interface. The former condition is verified only by mobile hosts that have joined the ad hoc network. The latter condition implies that the gateway can deliver traffic to that node only over multi-hop paths not traversing other gateways[3]. Thus, it is highly probable that the considered gateway is the default gateway selected by that ad hoc node. To illustrate how the proposed mechanism works, let us consider the network in Figure 3. When a node on the wired local LAN (e.g., node H) wants to send packets to an ad hoc node (e.g., node N), it assumes that the ad hoc node is on the same physical network. Hence, node H checks its ARP table for IP-MAC mapping and, if it is not present, it sends an ARP request. The gateway GW1 fulfills the previously defined conditions (i.e., node N's IP address has an entry in the GW1's routing table with a netmask $255.255.255.255$, which is related to its wireless interface), while GW2 does not. Consequently, only GW1 is allowed by the Proxy ARP server to answer with an ARP reply. This ARP reply will insert the mapping [node N's IP address - MAC address of GW1's wired interface] into the node H's ARP table. Thus, the packets sent from node H to node N will be delivered to GW1, which will forward them to node N. On the other hand, node N will reply to node H using GW1, as indicated by its routing table (see Table I).

There are some network configurations where asymmetric routing may occur, i.e., the forward path is different from the return path. For instance, let us consider the case in which node N is in radio visibility of two gateways GW1 and GW2. In this situation, the OLSR routing algorithm will randomly select one of these gateways as default gateway for node N. However, both gateways are allowed to send ARP replies for ARP requests issued by node H for the node N's IP address. In this case, the wired node H will update its ARP table using the information delivered in the last received ARP reply. Let us assume that GW1 is the default gateway for node N, but GW2 has sent the last ARP reply to node H. In this case, node H sends the traffic destined to node N to GW2, which routes it to node N. On the other hand, node N sends packets destined to node H to GW1, which forwards them to node H. It is important to note that asymmetric paths are not by themselves a problem. Indeed, both node N and H correctly receive and send their packets. In addition the asymmetric routing occurs only in symmetric topologies. Thus, it is reasonable to assume, in this local environment, that both paths are characterized by similar delays.

3) Mobility Support.: In general, solutions to support Internet connectivity for ad hoc networks, which are based on gateways, experience TCP-session breaks when the default route changes, depending on dynamics and mobility in the network. To avoid that TCP sessions break, in [12] it was proposed to replace default routes with explicit tunneling between the mobile nodes and the gateways. However, this complicates significantly the implementation and introduces relevant overheads. On the contrary, in our architecture the mobility is supported in a transparent way for the higher

[3]It is worth reminding that gateways are always interconnected using their wired interfaces. Hence, a route to reach a mobile node can traverse two gateways only if one of the link along the path is a wired link. In this case the farthest gateway will have the next-hop routing entry for that mobile node on its wired interface.

protocol layers. Indeed, the only effect of changing the default gateway for node N, is that the node N's outgoing traffic is routed towards the new gateway (e.g., GW2), while the initial gateway (e.g., GW1) continues to receive the incoming traffic and to forward it to node N. This results into asymmetric routing. However, this asymmetry can be easily removed by using an advanced feature of the ARP protocol. More precisely, when GW2 becomes aware that the next hop for the node N switches from its wired interface to its wireless interface, it generates a *Gratuitous ARP* on the wired interface for node N's IP address. This will update the ARP table in all of the wired hosts that have an old entry for the node N's IP address, which was mapped with the MAC address of GW1's wired interface. This action restores a symmetric path for the active packet flows destined to and/or originated from node N.

VI. Experimental Results

We have prototyped the core functionalities of our architecture. In particular, we have developed the software components described in Section V-B, concerning the support of Internet and Intranet connectivity for the ad hoc nodes. Currently, we are completing the implementation of the modifications to the DHCP Relay agents described in Section V-A. For these reasons, in the following we will show experimental results measuring the network performance with mobility and Internet access, while we left for further work the testing of the performance (such as address allocation latency and communication overheads) of the proposed node self-configuration scheme.

In our test-beds we have used *IBM R-50* laptops with *Intel Pro-Wireless 2200* as integrated wireless card. We have also used the OLSR_UniK implementation for Linux in version $0.4.8$ [19]. The installed Linux kernel distribution was $2.6.9$. The ad hoc nodes are connected via IEEE $802.11b$ wireless links, transmitting at the maximum rate of 11 Mbps. To generate the asymptotic UDP and TCP traffic during the experiments we used the *iperf* tool[4]. More precisely, the iperf server (termination of the traffic sessions) runs in a static host in the wired LAN, while iperf clients (originators of traffic sessions) have been set up on the mobile nodes. If not otherwise specified, the packet size is constant in all the experiments and the transport layer payload is equal to 1448 bytes. Differently from other studies [11], in which the network topology was only emulated by using the *IP-tables* feature of Linux, our experiments were conducted in realistic scenarios, with hosts located at the ground floor of the CNR building.

A. Performance Constraints of Internet Access

To measure the performance constraints in case of Internet access, we executed several experiments in the test-bed shown in Figure 4. The distances between the ad hoc nodes were set up in such a way to form a 4-hop chain topology with high-quality wireless links. The first set of experiments was conducted to evaluate the impact on the UDP and TCP

[4]`http://dast.nlanr.net/Projects/Iperf/`.

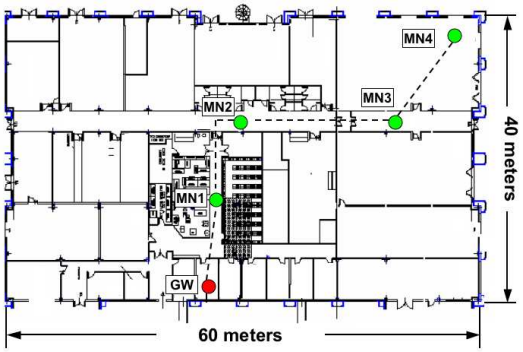


Fig. 4. Trial scenario for testing Internet access using a chain network.

throughput of the number of wireless hops traversed in the ad hoc network to reach the gateway. During these tests all the OLSR configuration parameters have been set up according to the default values indicated in the RFC specification [7]. Figure 5 and Figure 6 show the UDP and TCP throughput, respectively, obtained during a single experiment, as a function of the time and for different chain lengths. Several observations can be derived from the shown experimental results. First, we can note that the maximum UDP throughput is always greater than the maximum TCP throughput, for every network configuration. This is obviously due to the additional overheads introduced by the TCP return traffic, which consists of TCP ACK packets. In addition, as expected, the longer the route, the lower is the peak throughput achieved by the session flow (both TCP and UDP). The figures show also that, although the nodes are static, the throughput is not stable, but both UDP and TCP flows could be in a stalled condition for several seconds. An initial explanation of this route instability is that losses of routing control frames can induce the loss of valid routes. Indeed, the routing control frames are broadcast frames, which are neither acknowledged nor retransmitted, hence they are more vulnerable to collisions and channel errors than unicast frames. However, a careful analysis of the routing log files has pointed out another relevant condition that contributes to the route instability in our static network. Indeed, we discovered that the OLSR protocol implements an over pessimistic estimation of the link quality that may cause to consider as lost a link that is overloaded. More precisely, each node keeps updating a $link\_quality$ value for each neighbor interface. Every time an OLSR packet is lost $link\_quality = (1-\alpha) \cdot link\_quality$[5], while every time an OLSR packet is correctly received $link\_quality = (1-\alpha) \cdot link\_quality + \alpha$, where the α value is the smoothing factor of the estimator. The OLSR specification suggests as default configuration $\alpha = 0.5$. This implies that the $link\_quality$ value is halved after each OLSR packet loss. The $link\_quality$ parameter is used to estimate the link reliability, according to a procedure denoted as *link hysteresis* [7].

[5]To identify the loss of an OLSR packet two mechanisms are used: 1) tracking the sequence numbers of the received OLSR packets, or 2) monitoring OLSR packet receptions during an *HELLO* emission interval [7].

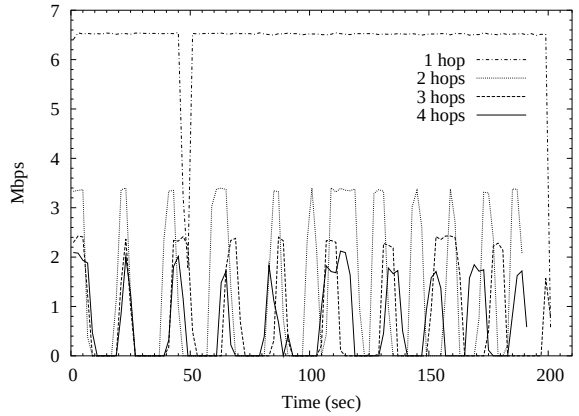


Fig. 5. Throughput of a single UDP flow for different chain lengths.

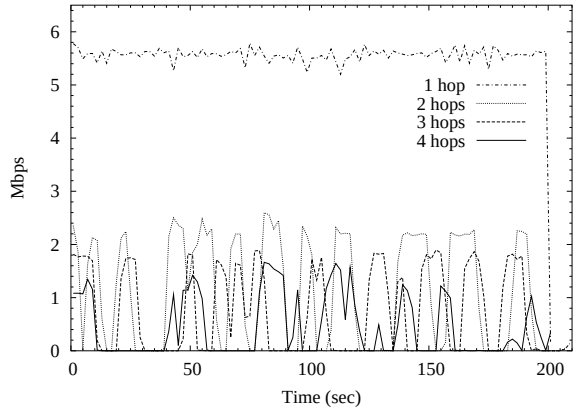


Fig. 6. Throughput of a single TCP flow for different chain lengths.

More precisely, the value of the $link\_quality$, is compared with two thresholds, called $HYST\_THRESHOLD\_LOW$ and $HYST\_THRESHOLD\_UP$. When $link\_quality < HYST\_THRESHOLD\_LOW$, the link is considered as *pending*, i.e., not established. A pending link is not completely dropped because the link information is still updated for each *HELLO* message received. However, a pending link is not a valid link when computing routing tables. In addition, a pending link can be considered again as established only when $link\_quality > HYST\_THRESHOLD\_UP$. The OLSR specification suggests as default configuration $HYST\_THRESHOLD\_LOW = 0.3$ and $HYST\_THRESHOLD\_UP = 0.8$. According to these values and to the scaling factor α, even a perfect link (i.e., a link with $link\_quality = 1$) will be purged from the routing tables when two consecutive OLSR packets are lost. We argue that the standard setting of the hysteresis parameters introduces a critical instability in the routing tables, because it is not infrequent to loose broadcast packets (as the OLSR packets are) when the channel is overloaded.

To verify our claim we have carried out a second set of experiments in the same network configuration depicted in Figure 4, disabling the OLSR hysteresis process. To provide

TABLE III
UDP THROUGHPUT IN A CHAIN NETWORK, WITH AND WITHOUT HYSTERESIS.

	UDP		
	HYST	NO HYST	
1 *hop*	6.124Mbps (304Kbps)	6.363Mbps (393Kbps)	+4%
2 *hops*	1.252Mbps (55Kbps)	2.501Mbps (57Kbps)	+100%
3 *hops*	700.4Kbps (60Kbps)	1.306Mbps (87Kbps)	+86%
4 *hops*	520.6Kbps (54Kbps)	1.141Mbps (56Kbps)	+119%

TABLE IV
TCP THROUGHPUT IN A CHAIN NETWORK, WITH AND WITHOUT HYSTERESIS.

	TCP		
	HYST	NO HYST	
1 *hop*	5.184Mbps (335Kbps)	5.172Mbps (393Kbps)	≈=
2 *hops*	956.1Kbps (123Kbps)	1.517Mbps (57Kbps)	+58%
3 *hops*	638.1Kbps (149Kbps)	891.7Kbps (77Kbps)	+39%
4 *hops*	345.9Kbps (47Kbps)	631.2Kbps (74Kbps)	+82%

statistically correct results, we have replicated each experiment five times. Tables III and Table IV show the average and standard deviation (in parenthesis) values of the measured throughputs for the UDP and TCP case, respectively. From the results we observe that the throughput performances are significantly improved, with the improvement for a 4-hop chain reaching 119% in the UDP case and 82% in the TCP case. The study of routing table logs clearly indicates that these throughput increases are due to an improvement in the route stability with less frequent declarations of link drops due to erroneous estimations of links' reliability. It is worth pointing out that this issue has not been identified in previous experimental studies because either the multi-hop communications where only emulated [12], or the channel was loaded with low-intensity *ping* traffic [20].

In addition to the hysteresis process, the OLSR protocol employs several other mechanisms, as the link sensing, neighbor detection and topology discovery, which significantly affect the route stability. Indeed, recent works [20], [21] have investigated how the setting of the classical OLSR routing parameters may affect the network performances. However, these works have specifically focused on the time required for route recalculation after a link drop due to node mobility. On the contrary, to conclude this section we will analyze the impact of different OLSR parameter settings on the performance limits of Internet access in static network configurations. More precisely, each OLSR packet, and the information it delivers, has a fixed validity time. For instance, the information provided in a *HELLO* message is considered valid for a *NEIGHB_HOLD_TIME*. This implies that a node detects a link loss with a neighbor from the lack of *HELLO* messages during a *NEIGHB_HOLD_TIME*. A similar check is performed for the *TC* messages, whose validity time is *TOP_HOLD_TIME*, and for the *HNA* messages, whose validity

time is *HNA_HOLD_TIME*. A possible strategy to avoid that links and routes are dropped from the routing tables because the related information has not been refreshed within the corresponding timeout, is to increase the frequency used to generate OLSR packets. This may increase the probability that at least one new OLSR packet is received before its validity time expires. The drawback of this approach is that the more frequent the OLSR protocol generates control messages, the higher is the routing overheads. To quantify the trade-off between routing overhead increases and route stability improvements, and how this impacts network performance, we have carried out a set of experiments in a 3-hop chain using the OLSR parameter settings shown in Table V. As listed in the table, we compare the default parameter setting with disabled hysteresis (*set1*) with the cases in which the frequency of OLSR packet generations is two times (*set2*) and four times (*set3*) higher, while the validity times are kept constant.

TABLE V
OLSR PARAMETER CONFIGURATIONS.

OLSR parameters	*set1*	*set2*	*set3*	*default*
HELLO_INTERVAL(s)	2	1	0.5	2
NEIGHB_HOLD_TIME(s)	6	6	6	6
TC_INTERVAL(s)	5	2.5	1.25	5
TOP_HOLD_TIME(s)	15	15	15	15
HNA_INTERVAL(s)	5	2.5	1.25	5
HNA_HOLD_TIME(s)	15	15	15	15
Hysteresis	no	no	no	yes

TABLE VI
UDP AND TCP THROUGHPUTS IN A 3-HOP CHAIN NETWORK FOR DIFFERENT OLSR PARAMETER SETTINGS.

Parameter Setting	UDP	TCP
default	700.4Kbps (60Kbps)	638.1Kbps (149Kbps)
set1	1.306Mbps (87Kbps)	838.5Kbps (79Kbps)
set2	1.605Mbps (76Kbps)	1.020Mbps (105Kbps)
set3	1.84Mbps (106Kbps)	1.306Mbps (56Kbps)

The experimental results obtained by replicating five times the throughout measurements for UDP and TCP traffic are listed in Table VI, in which the average throughput and its standard deviation (in parenthesis) are reported. The shown results indicate that increasing the frequency the OLSR packets are generated by a factor of four, and maintaining the default validity times, it is possible to improve the average throughput of 40% in the UDP case, and of 55% in the TCP case. We have analyzed the routing table logs generated during the trials, and again we have observed that the throughput increases are due to an improvement in route stability. On the other hand, the increase of routing overheads has a negligible impact on the throughput performance.

In summary, our experimental study indicates that the network performance of Internet access in static configurations can be significantly enhanced (in some cases we have more than doubled the measured throughputs) by properly setting the OLSR parameters such as to improve route stability.

B. Performance Constraints with Mobility

To test the mobility support in a multi-homed network configuration we considered the network layout illustrated in Figure 7. In our experiments, node MN2 alternates between position P1 and position P2. More precisely, it starts in position P1, where it is in radio visibility of node MN1. After 50 seconds it moves in position P2, where it is in radio visibility of node MN3. The time needed for moving from P1 to P2 is 20 seconds. After other 50 seconds, host MN2 goes back to position P1. This mobility patterns is periodically repeated throughout the test. The *HNA* messages from GW1 and GW2 form default routes to the external network on a short-hop basis. Hence, while connected to MN1, the node MN2 uses GW1 as default gateway. On the contrary, when connected to node MN3, the routes are recalculated and MN2 uses GW2 as default gateway. The new default gateway GW2 will also begin to act as Proxy ARP for the mobile node. The return traffic will be consistently routed through the new gateway as soon as either a new ARP request for the MN2's IP address is issued by the external host, or the gateway GW2 sends a Gratuitous ARP; otherwise it will continue to arrive at the GW1 (see Section V-B.3 for the details).

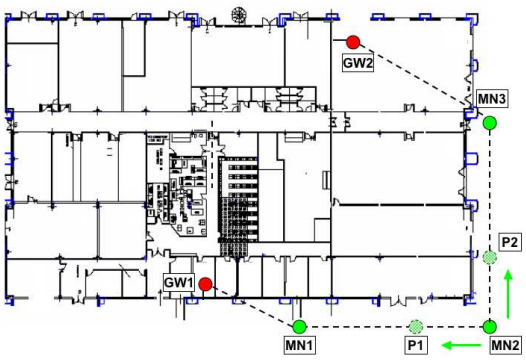


Fig. 7. Trial scenario for testing mobility support.

Figure 8 shows the TCP throughput achieved by MN2 during a mobility test. We compare these results against the throughput measured when node MN2 is fixed in position P1. During both experiments, the hysteresis process was disabled and the other OLSR parameters were set up according to the default values indicated in the RFC specification [7]. The shown results confirm that the TCP session does not break when node moves. The major effect of node mobility is to introduce holes in the TCP traffic due to the time needed to recalculate the new routes to reach the default gateway. In the considered case of "soft" handoff, i.e., the mobile nodes is in radio visibility of both node MN1 and node MN3 when changing position, we measured up to 20 seconds for recomputing a consistent routing table in node MN2. It is worth noting that in similar experiments conducted in [11], the throughput of mobile node was approximately 30% lower

when mobile node changed position. This was expected because the TCP-session continuity was ensured at the cost of using IP tunneling that introduces significant additional overheads. On the contrary our solution is very efficient and lightweight, because it operates directly at the data link layer.

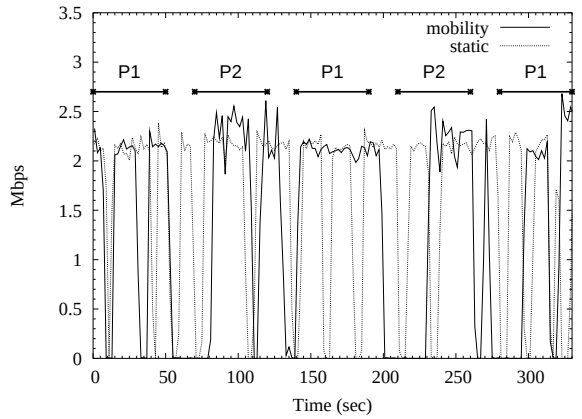


Fig. 8. Throughput of a single TCP flow with node mobility.

VII. Concluding Remarks

In this paper, we have presented a practical architecture to logically extend traditional wired LANs using multi-hop ad hoc networking technologies. Our proposed architecture provides ad hoc node self-configuration and both Intranet and Internet connectivity in a way that is transparent to the wired nodes, i.e., without requiring changes in the pre-existing wired LAN. In addition, by locating our architecture below the IP level, we have designed a lightweight and efficient ad hoc support framework, which is easy to be implemented and introduces minimal overheads.

We have prototyped the proposed architecture to test its functionalities. The shown experimental results indicates that: i) the network performance of Internet access in static configurations can be significantly enhanced (in some cases we have more than doubled the measured throughputs) by properly setting the OLSR parameters such as to improve route stability; and ii) the continuity of TCP sessions during node mobility is achieved without requiring additional overheads.

Several possible extensions of this architecture are currently under investigation. Firstly, different ad hoc routing protocols could be used in separated ad hoc components, either proactive or reactive. The interactions between gateways based on different solutions is not a trivial problem to handle. Moreover, providing a full IP compatibility requires that the ad hoc network should support not only unicast routing but also multicast and other advanced IP functionalities. We are currently investigating how our solution could be extended to provide the complete IP support.

VIII. Acknowledgements

The authors are thankful to the anonymous reviewers for the useful comments in improving the quality of the paper. This work was partially funded by the Italian Ministry for Education and Scientific Research (MIUR) in the framework of the FIRB-VICOM project, and by the Information Society Technologies program of the European Commission under the IST-2001-38113 MobileMAN project.

References

[1] R. Bruno, M. Conti, and E. Gregori, "Mesh Networks: Commodity Multihop Ad Hoc Networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123–131, March 2005.

[2] P. Srisuresh and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663, August 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2663.txt

[3] C. Perkins, "IP Mobility Support for IPv4," RFC 3344, August 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3344.txt

[4] A. Acharya, A. Misra, and S. Bansal, "A Label-switching Packet Forwarding Architecture for Multi-hop Wireless LANs," in *Proc. of ACM WoWMoM 2002*, Atlanta, Georgia, USA, September, 28 2002, pp. 33–40.

[5] C. Tschuding, R. Gold, O. Rensfelt, and O. Wibling, "LUNAR: a Lightweight Underlay Network Ad-hoc Routing Protocol and Implementation," in *Proc. of NEW2AN'04*, St. Petersburg, Russia, February, 2–6 2004.

[6] R. Draves, J. Padhye, and B. Zill, "The architecture of the Link Quality Source Routing Protocol." Microsoft Research, Tech. Rep. MSR-TR-2004-57, 2004.

[7] T. Clausen and P. Jaquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626, October 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[8] R. Ogier, F. Templin, and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)," RFC 3684, February 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3684.txt

[9] M. Benzaid, P. Minet, K. Al Agha, C. Adjih, and G. Allard, "Integration of Mobile-IP and OLSR for a Universal Mobility," *Wireless Networks*, vol. 10, no. 4, pp. 377–388, July 2004.

[10] U. Jönsson, F. Alriksson, T. Larsson, P. Johansson, and G. Maguire Jr., "MIPMANET - Mobile IP for Mobile Ad Hoc Networks," in *Proc. of MobiHoc 2000*, Boston, MA, USA, August, 11 2000, pp. 75–85.

[11] P. Engelstad, A. Tønnesen, A. Hafslund, and G. Egeland, "Internet Connectivity for Multi-Homed Proactive Ad Hoc Networks," in *Proc. of IEEE ICC'2004*, vol. 7, Paris, France, June, 20–24 2004, pp. 4050–4056.

[12] P. Engelstad and G. Egeland, "NAT-based Internet Connectivity for On Demand MANETs," in *Proc. of WONS 2004*, Madonna di Campiglio, Italy, January, 18–23 2004, pp. 4050–4056.

[13] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131, March 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2131.txt

[14] N. Vaidya, "Weak Duplicate Address Detection in Mobile Ad Hoc Networks," in *Proc. of ACM MobiHoc 2002*, Lausanne, Switzerland, June, 9–11 2002, pp. 206–216.

[15] S. Nesargi and R. Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc NEtwork," in *Proc. of INFOCOM 2002*, vol. 2, New York, NY, June, 23–27 2002, pp. 1059–1068.

[16] K. Weniger and M. Zitterbart, "Address Autoconfiguration on Mobile Ad Hoc Networks: Current Approaches and Future Directions," *IEEE Network*, vol. 18, no. 4, pp. 6–11, July/August 2004.

[17] S. Carl-Mitchell and J. Quarterman, "Using ARP to Implement Transparent Subnet Gateways," RFC 1027, October 1987. [Online]. Available: http://www.ietf.org/rfc/rfc1027.txt

[18] D. Plummer, "An Ethernet Address Resolution Protocol," RFC 826, November 1982. [Online]. Available: http://www.ietf.org/rfc/rfc0826.txt

[19] A. Tønnesen. (2004, December) Implementation of the OLSR specification (OLSR_UniK). Version 0.4.8. University of Oslo. [Online]. Available: http://www.olsr.org/

[20] E. Borgia, "Experimental evaluation of ad hoc routing protocols," in *Proc. of IEEE PerCom 2005 Workshops*, Kauai Island, Hawaii, March, 8–12 2005.

[21] C. Gomez, D. Garcia, and J. Paradells, "Improving Performance of a Real Ad-hoc Network by Tuning OLSR Parameters," in *Proc. of IEEE ISCC 2005*, Cartagena, Spain, June, 27–30 2005, pp. 16–21.

Model Based Protocol Fusion for MANET-Internet Integration

Christophe Jelger
Fraunhofer Institute FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
Christophe.Jelger@fokus.fraunhofer.de

Christian Tschudin
Computer Networks Research Group
University of Basel, Bernoullistrasse 16,
CH-4056 Basel, Switzerland
Christian.Tschudin@unibas.ch

***Abstract*— With the wide adoption of wireless communication technologies, the current networking design of the Internet architecture has shown some limitations. Restricted by inherent layering constraints, valuable networking information cannot flow freely inside the network stack and potential operational optimizations are impossible to achieve. To overcome these limitations, we extend the current trend of cross-layer approaches with a framework called *underlay protocol fusion*: the basic building blocks of Internet functionality are factorized out and merged in a function pool where information sharing and operational optimizations are performed.**

To illustrate our approach, we present LUNARng (LUNAR next generation). It is a fully distributed underlay protocol designed for the Internet integration of wireless ad hoc networks (MANETs) where fundamental services such as name resolution, address autoconfiguration, and IPv4/IPv6 routing are transparently available whether the MANET is connected or not to the Internet. Internet integration refers here to the ability to *insert/remove* a MANET *into/from* the logical organization of the Internet without any loss of functionality. Moreover by using *protocol models*, the underlay nature of LUNARng allows to optimally merge (with respect to the multi-hop nature of MANETs) network operations which are traditionally carried out at different layers of the protocol stack.

***Index Terms*— Underlay MANET, Internet Integration, Protocol Fusion.**

I. Introduction

A. Internet integration

The design of the Internet is based on a thirty-years old layering approach which aimed at factoring out functionality. Networking concepts were sought which are able to stretch from local scale to global size, from slow links to highspeed trunks, from PDAs to supercomputers. In spite of its slow evolution and monolithical design, the "canonical set" of protocols collected in the Internet-Suite has done a surprisingly good job during the last decade. Today however, the limitations of this one-size-fits-all approach have become visible especially with the advent of wireless communications.

These limitations can be linked to the incapacity of the Internet to scale in a functional way at the networking layer. It has evolved through a patch style which has not added variety: additions were made in a stealth way and have not dared to radically change or extend the core of IP forwarding. We refer here to the introduction of the hidden routing hierarchy and mechanisms of AS, CIDR or MPLS as well as other less successful projects (in terms of large-scale deployment) like RSVP, IP Multicast, and Mobile IP. Due to the end-to-end principle, the place where the Internet has envisaged and endorsed functional scaling, that is the possibility to freely add arbitrary customized functionality, is the application layer. This is where remarkable breakthroughs have been achieved and variety was obtained: DNS, eMail, Web, VPN, VoIP and P2P are the highlights to mention here.

The low flexibility of the current Internet protocol suite is particularly striking when considering mobile wireless ad hoc networks (MANETs). The inherent distributed and infrastructure-less nature of MANETs has indeed highlighted how fundamental services of the Internet rely on a centralized client-server model. That is, the absence of basic services such as name resolution or address allocation does not strictly prevent networking, but it strongly restrains the adoption of wireless ad hoc networking as a plug-and-play technology for the masses. Therefore the ability of an autonomous MANET to exhibit Internet-like functionalities (while not being connected to the Internet) is one facet of Internet integration: users should not experience a loss of commodity other than the loss of global connectivity.

A complementing aspect is the seamless integration of (mesh) MANETs with the logical (e.g. global addressing) and operational (e.g. name resolution) organization of the Internet. This property is the second facet of Internet integration: the ability for a MANET to adapt its internal behavior in order to *insert* itself into a larger organization such as the Internet.

B. Underlay design for MANETs

As stated earlier, the strict layering approach of the existing legacy TCP/IP model is slowing down or even preventing innovative functionalities to appear at the network layer. It is commonly agreed that more inter-layer coordination is needed in wireless networks when considering on one side the network layer, and on the other side the physical and the link layers [1]. However, more coordination is also required among higher

This work was carried out while Christophe Jelger was being sponsored by an ERCIM fellowship during his stay in the Computer Networks Research Group at the University of Basel. He is now with Fraunhofer Fokus, also sponsored by an ERCIM fellowship.

layer protocols. For example, we already demonstrated in [2] how a single DNS name resolution request procedure can gather enough *cross-protocol* data to fulfill the tasks of link-layer resolution and path setup.

In order to achieve such optimizations, we introduce an underlay shim that performs *protocol fusion* based on *protocol models*. The goal is first to gather the previously isolated information provided by different task-specific protocols of the network stack, and second to optimize the operation of these protocols by anticipating their needs. In contrast to more classical cross-layer techniques which provide hints and triggers between layers, we use an underlay located between the IP and Ethernet layers in order to have full control over the data coming in and out of a node. Tasks that were previously carried out by remote servers at different layers are now performed at layer 2.5 (i.e. hence the name *underlay*), and the historical barriers between the somehow isolated protocols involved at the network layer are suppressed. As a result, effective optimizations can now be achieved. Moreover, to realize the two facets of Internet integration, the functionalities provided at layer 2.5 are activated on-demand when the MANET is not connected to the Internet, or they can be bypassed depending if a given service is provided by the infrastructure-based network which provides the global connectivity.

The paper is organised as follows. In the next section, we situate our approach with respect to traditional cross-layer schemes and we introduce our underlay technique. We then briefly introduce LUNARng and summarize the features it provides. We then describe the mechanisms that we use to perform the Internet integration of wireless ad hoc networks, and we also detail the concept of protocol fusion. We also present some implementation details of our approach which has been successfully validated and deployed on a real testbed. Finally, we conclude with a discussion of future open research challenges.

II. Cross-layering vs. protocol fusion

A. Traditional cross-layer design

Cross-layer design is an active field of research which so far has not looked into Internet integration issues. Instead, the focus is on *wireless* networking, mainly because many protocols and services of the *wired* Internet are inefficient in the presence of unreliable wireless links and unpredictable topological changes. Without cross-layer design, the existing layer boundaries unfortunately prevent the development of potential optimizations[1]. Since the differences between wired and wireless networks lie in the physical and link layers, i.e. the layers located below the IP layer, a large majority of cross-layer techniques concentrates on providing lower-layer feedback to the network layer [1] (e.g. to notify link-layer events such as layer 2 handovers) and, to a smaller extent, to the transport layer (e.g. to achieve TCP performance optimizations via wireless-specific fine-tuning of TCP parameters). We denote such approaches as lower-layer feedback techniques, and illustrate them by Fig. 1 where valuable information is passed from the lower-layers to the higher layers, which can subsequently optimize their operation. The main advantage of this approach is that it becomes possible to re-design a given protocol such that it specifically reacts to some lower-layer signals, i.e. the protocol becomes aware of what happens in lower-layers. However, such lower-layer awareness does not come without a cost: it requires code modifications inside the protocol stack which can restrict the possible wide-scale adoption of such changes. As a consequence, most cross-layer schemes remain research prototypes. It is also worth mentioning that cross-layer designs using an *information-bus* (i.e. a transversal layer that receives/sends specific feedback from/to all layers via specific function calls) unfortunately suffer from the same implementation and deployment restrictions.

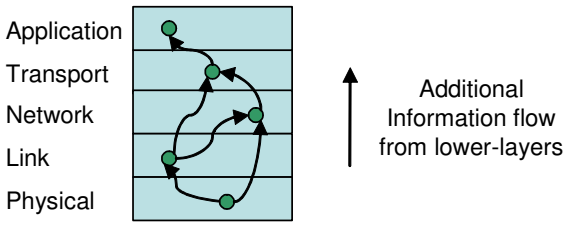


Fig. 1. Lower-layer feedback

B. Case study: the failure of name resolution

In the Internet, name resolution is performed via the Domain Name System (DNS) which relies on a hierarchy of servers distributed around the world. One issue is that dynamic name resolution in traditional IP networks assumes that there exists a reachable DNS server at all time: the whole operation of name resolution collapses if no server is available. All existing operating systems do not even try to send a name request if no DNS server is configured in the system: if a node is not configured with a DNS server address, it simply assumes that dynamic name resolution is not available. The implementation of a DNS-compatible name resolution system in a MANET is therefore challenging in many ways.

Actually, a *natural* way of performing name resolution in a MANET is to use a decentralized approach in which a node of the MANET replies to a broadcasted name request for which it is the target. Different flavors [2][3][4] of such an approach can be found in the literature. Although the operation of distributed name resolution resembles the route discovery procedure of a reactive routing protocol, it is more difficult to implement than routing since the DNS operation is *hard coded* in current operating systems and applications. That is, by default, a node configured with a DNS server address sends its unicast DNS request messages via the network interface towards the server. In a MANET, this procedure becomes irrelevant and it should be replaced with a MANET specific mechanism.

[1]One can note that the *restricted-scope* of competencies of standardization bodies (i.e. IEEE vs. IETF) also restricts the design and potential outcomes of cross-layer optimizations.

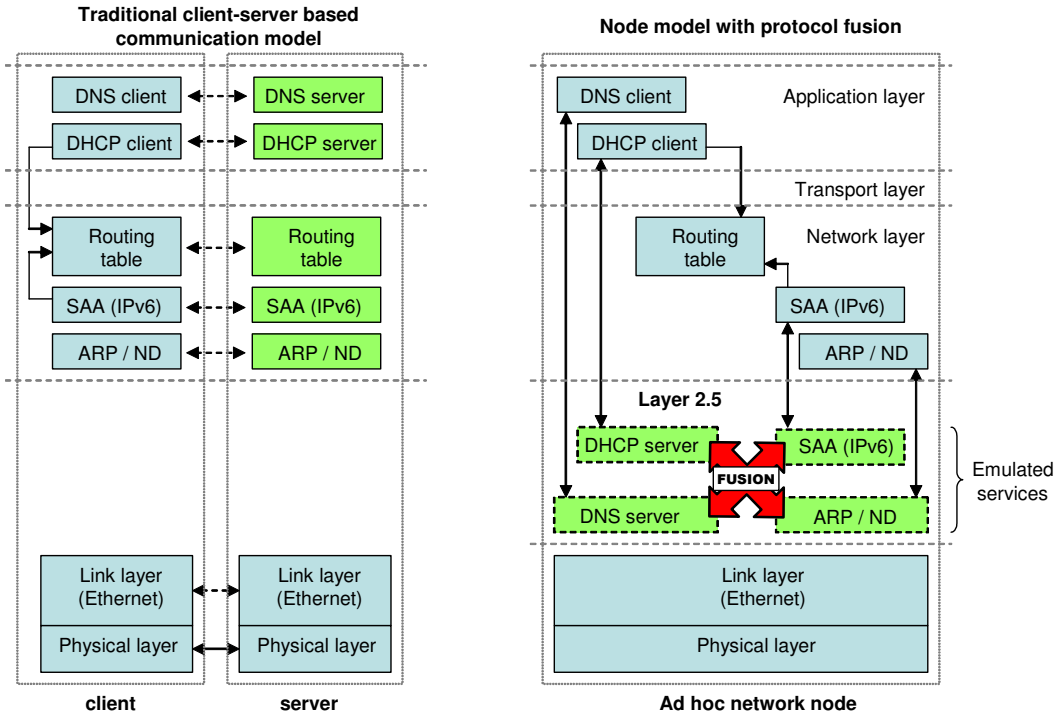


Fig. 2. Protocol emulation and fusion

Moreover, a name resolution scheme for MANETs should not prevent a node from resolving names in the classical way when the MANET is connected to an infrastructure-based network. In particular, the existing APIs and related protocols should remain unchanged since it is not conceivable to modify the huge amount of existing applications such as web browsers and email clients.

C. Internet integration (part I): service emulation with models

As introduced earlier, our approach relies on an underlay protocol located below the IP layer and above the Ethernet layer. Being located below IP, the underlay protocol is aware of all traffic coming in and out of a node. By manipulating the stream of messages passing by, it becomes possible to perform some *cross-protocol* optimizations. By cross-protocol optimizations we mean that, among other options it is possible to merge the operation of independent protocols by anticipating the needs of a given protocol: the global behavior of a TCP/IP-based protocol stack is indeed well known. For example, with a reactive routing protocol for wireless ad hoc networks (such as AODV [5] and DSR [6]), a successful route request (RREQ) procedure will always be followed by a link-layer resolution request for the next hop node towards the destination. One can therefore design the route request procedure such that it also performs the anticipated link-layer address resolution. A main advantage of using an underlay protocol is that it requires no modifications to the existing protocol stack. Furthermore, protocol fusion is invisible to legacy layers: it is thus possible to *fool* the higher layers by sending appropriate control messages to the protocols located in both the network and the application layers. It thus becomes possible to build *protocol models* which mimic the behavior of some specific network functionality in order to provide Internet-like services while the MANET is not connected to the Internet. We define this property of Internet integration as *service emulation*. One can note that the underlay can also be used to hide the multi-hop nature of a wireless ad hoc network such that the IP stack believes that the node is connected to a classical IP subnet on a single layer 2 link.

The concept of protocol emulation and fusion is illustrated by Fig. 2. The left side of the figure shows the client-server model used in traditional Internet-like networking. Fundamental services such as domain name resolution (DNS) and address autoconfiguration (DHCP for IPv4/IPv6, and SAA [7] for IPv6) fully rely on the presence of a dedicated server. Other protocols such as ARP and ND [8] are not based on the client-server model but for optimization purposes (as described later) their operation is also preempted by the underlay. In the right part of Fig. 2, we illustrate the underlay-based models located at layer 2.5. In order to overcome the absence of legacy servers, models of the basic Internet services are implemented in the underlay: these *emulated* functionalities are activated on-demand when required. Moreover, at the underlay level it is now possible to optimize the operation of protocols which were previously opaque to eachother. With this protocol fusion, network operations become optimized to the topological and operational properties of wireless ad hoc networks.

III. LUNARng

To explore and demonstrate the feasibility of underlay fusion, protocol emulation, and Internet integration, we have extended the original operation of LUNAR [9], i.e. a reactive routing protocol initially designed to back up the development of network pointers [10]. As for the features provided by the protocol fusion in the underlay, our next generation LUNAR (LUNARng) combines IPv4 and IPv6 path setup, link-layer address resolution (ARP/ND), and name resolution in a single request/reply operation optimized for distributed wireless ad hoc environments. Moreover, thanks to the use of network pointers as the basic forwarding abstraction, an IPv6 multi-hop data path can include nodes which are only IPv4 enabled (and vice-versa).

The two facets of Internet integration are also covered, since LUNARng provides Internet-like services via protocol emulation when the MANET is autonomous, and Internet adaptation via coherent global addressing, routing, and name resolution when the MANET is connected to the Internet. Moreover Internet integration is transparent to the MANET users, in the sense that they only witness the appearance or loss of global connectivity.

In practice, LUNAR is implemented as a Linux kernel module[2] that can be dynamically loaded on a host and which requires no single modification to the Linux kernel code. LUNAR positions itself between the IP and Ethernet layers (actually just above the wireless device driver) and creates a subnet illusion with respect to the IP stack. Upon startup, the LUNAR module creates a virtual network interface that is internally linked with the real wireless interface connected to the MANET. Hence, all traffic that flows via the virtual interface is seen by the LUNAR module which can specifically react to particular messages.

IV. Internet integration implemented

In this section, we describe the mechanisms developed in LUNARng in order to perform Internet integration and optimize the operation of Internet-like protocols with respect to distributed wireless networking.

A. Filling expectations and building models

As stated previously, the basic steps and behavior of a communication startup with the TCP/IP protocol stack is well known. In a wired network, the initiation of a communication usually conforms to the following steps (we assume that each step is successful):

1. The user specifies the name of the host s/he wishes to communicate with,
2. The IP stack triggers an ARP/ND (IPv4 Address Resolution Protocol or IPv6 Neighbor Discovery) request in order to resolve the link-layer address of the next hop towards the DNS server (or of the DNS server itself),
3. The system sends a DNS request to resolve the target host name,
4. The IP stack sends an ARP/ND request in order to resolve the link-layer address of the next hop towards the target (or of the target itself).

In a MANET that uses a reactive routing protocol, if we assume that a node is configured with a DNS server address, and if we assume that there exists a DNS server in the MANET, the following steps are executed:

1. The user specifies the name of the host s/he wishes to communicate with,
2. The MANET routing module triggers a route request (RREQ) procedure to find a path to the DNS server,
3. The IP stack triggers an ARP/ND request in order to resolve the link-layer address of the next hop towards the DNS server (or of the server itself),
4. The system sends a DNS request to resolve the IP address of the target host name,
5. The MANET routing module triggers a RREQ/RREP procedure to find a path to the IP address of the target host,
6. The IP stack sends an ARP/ND request in order to resolve the link-layer address of the next hop towards the target (or of the target itself).

It is clear that the specificities of MANETs already increase the total overhead because routes have to be discovered on-demand. Moreover, we assumed here that there existed a DNS server, but this assumption will usually be false in real infrastructure-less MANETs. Hence to resolve the specific issues introduced by the distributed and autonomous nature of wireless ad hoc networks, we introduce an optimized scheme which specifically considers and addresses the particular features of MANETs.

B. Revisiting route requests

The key element of our underlay approach is that in a MANET, one can trigger the route request procedure with the **name** of the destination host rather than with its IP address. At the same time, if the RREQ procedure can gather enough information, the number of required steps can now be greatly reduced:

1. The user specifies the name of the host s/he wishes to communicate with,
2. The MANET routing module triggers a route request RREQ/RREP procedure to find a path to the specified target host name. The RREP eventually contains the target IP address(es) and the link-layer address of the next hop towards the target.

Once the originating host receives this information bundle, it knows all relevant details to answer subsequent information requests (ARP) internally, and the communication can start immediately. The path discovery procedure is thus triggered by the name request message, i.e. an application layer protocol. The route is then discovered (network layer), and the link-layer address is resolved at the same time (network and link layers). Our underlay approach perfectly suits to the above optimization since we can capture the initial DNS request and translate it to an appropriate RREQ message: with a single RREQ/RREP procedure, we have performed name resolution, path setup, and link-layer address resolution.

[2]Available at http://core.it.uu.se/adhoc - The Uppsala/Basel Ad-Hoc Implementation Portal.

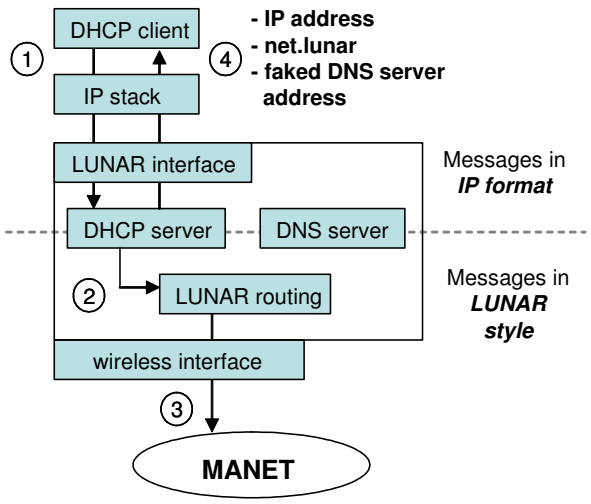


Fig. 3. LUNAR DHCP operation

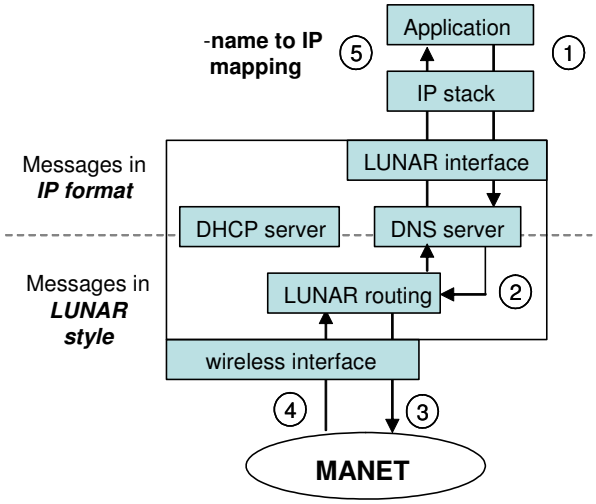


Fig. 4. LUNAR DNS operation

C. Internet integration (part II): Internet adaptation

On top of the merging of resolution requests, LUNARng supports address autoconfiguration by implementing a virtual DHCP server which assigns the IP address to the virtual interface, should the user want to automatically configure this interface via DHCP. This mechanism is illustrated by Fig. 3. In step 1 of Fig. 3, a DHCP client sends a request towards the LUNAR interface. This request is intercepted by the DHCP engine of LUNAR which randomly chooses an address within a pre-defined LUNAR subnet (e.g. 192.168.42.0/24), and which then checks the uniqueness of this address by trying to build a path towards this address (steps 2 and 3). If the path setup fails (i.e. indicating that the address is not used), a faked DHCP message is sent to the dhcp client application (step 4). This message includes the IP address to be used on this interface, the *net.lunar* domain name, and the address of a faked DNS server reachable via the LUNAR interface.

We also use a similar mechanism in order to perform IPv6 stateless address autoconfiguration (SAA [7]). LUNAR intercepts router solicitation (RS) messages sent by the IP stack, and returns back a faked router advertisement (RA) message which contains a MANET-global prefix[3].

Moreover, in order to perform Internet adaptation when the MANET is connected to the world-wide network, LUNARng also supports global address autoconfiguration based on *prefix continuity* [11]. That is when there exists a gateway to the Internet, LUNARng can coherently distribute a topologically correct and globally routeable prefix to the nodes of the MANET. The subnet illusion is maintained, and IPv6 multi-homing is also possible (if multiple gateways are present). Moreover, DNS requests for targets which are not inside the MANET are forwarded to the gateway: this is made possible since we introduce a virtual namespace as described in subsection IV-E.

[3]Since IPv6 scope-local addresses have been deprecated, we currently use unique local IPv6 unicast addresses (FC00::/7) as MANET-local addresses. See the recently standardized IETF RFC-4193 for details.

D. DNS operation

The interception of a DNS request is illustrated by Fig. 4. In step 1 of Fig. 4, an application triggers the sending of a DNS request that is intercepted by the LUNAR DNS engine. This triggers a route request procedure which uses the name of the target to identify the expected destination (steps 2 and 3). When the target discovers its name in the route request message, it sends back a route reply message which contains its IP address (steps 4 and 5). Note that this message also contains the MAC address of the next hop node towards the target destination: the node which triggered the name resolution request therefore also implicitly performs the link-layer resolution usually carried out by the ARP and ND protocols. The LUNAR module can then send back a classical DNS reply message to the application which eventually learns the IP address of the target. At that point, the network path is already established and the link-layer address of the next hop node is already known. When the IP stack subsequently issues an ARP or neighbor discovery (ND) request, the LUNAR module, which also intercepts these messages, can reply immediately without sending out any messages on the network.

E. net.lunar

The *net.lunar* domain name is configured at startup by the LUNAR module as being the default domain of a MANET node. Hence, a hostname request (i.e. not fully qualified) is transformed by the operating system into a net.lunar request which is recognized by the LUNAR module as being a MANET name resolution. In this way it becomes possible to identify a simple hostname lookup within the MANET if the user/application only specifies a hostname (e.g. the request for *cjelger* becomes a request for *cjelger.net.lunar*).

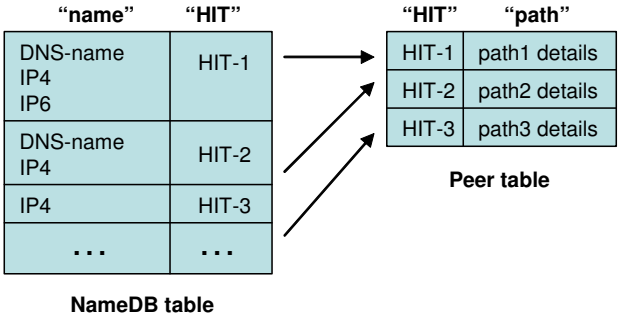


Fig. 5. NameDB and peer tables

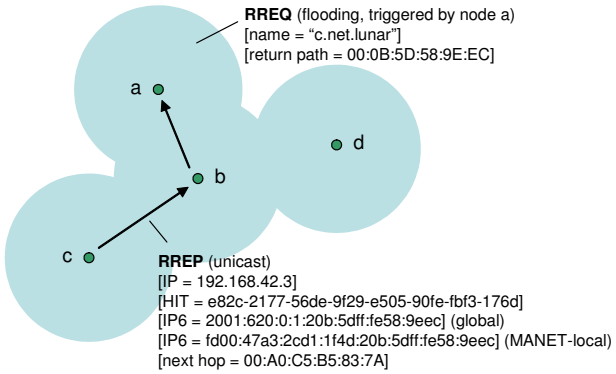


Fig. 6. RREQ/RREP procedure

In other words, a user can easily express its desire to trigger a name resolution inside the MANET. In contrast to hostname requests, FQDN (fully qualified domain name) requests (e.g. informatik.unibas.ch) are left unchanged by the operating system and the LUNAR module will recognize them as being a request for a host located outside the MANET. If the MANET is connected to the Internet via a gateway, non *net.lunar* DNS requests can be forwarded to the gateway which will potentially contact a traditional DNS server.

F. "Name" cache and path discovery

In order to avoid unnecessary path discovery procedures, each LUNAR node maintains a "name" cache, the so called NameDB (DataBase) table as shown by Fig. 5. This table contains all the known identifiers of a given correspondant (also called a *peer*): a DNS-name, one IPv4 and possibly multiple IPv6 addresses, and a host identifier tag or HIT inspired from the Host Identity Protocol (HIP [12]). As a HIT we use a random 128-bit string which becomes the entity which LUNAR uses to re-establish paths to a peer. That is, with either name resolution or plain ARP or ND resolution, we bind a peer's name and address to its HIT. All subsequent path lookup will be carried out with the HIT: as long as our name cache contains an entry for a peer, we will address this peer using its HIT. In order to populate the NameDB table, we use the LUNAR route discovery procedure to obtain the identifiers of a given target, as illustrated by Fig. 6.

Moreover, each node can gratuitously populate its NameDB table by overhearing the information contained in the RREQ messages it forwards. Also, a quiet node which does not send RREQ messages can also send unsollicited HELLO messages in order to notify the network about its existence. A user can thus learn the names of other computers connected to the MANET, since a summary of the information contained in the NameDB table is available via the Linux `/proc/net/lunar` file, illustrated below by Fig. 7. On this figure we can see the *emulated* DNS server peer, the localhost *baobab* which is IPv4 and IPv6 enabled, the peer *mango* for which a path (IPv4 and IPv6) is active (peer is resolved), the IPv6-only gateway *banana*, and the peer *cactus* for which a

```
# lunar (32 table entries, 4 peers)

FLAGS: N=name, H=hit, 4=IPv4, 6=IPv6,
       r=resolved, s=search, G=gateway

baobab.net.lunar    NH46..   -> Localhost
dns.net.lunar       N.4.r.   -> No lifetime
mango.net.lunar     NH46r.   -> Last heard 6s ago
banana.net.lunar    NH.6rG   -> Last heard 13s ago
cactus.net.lunar    N...s.   -> No lifetime
```

Fig. 7. The `proc/net/lunar` file.

RREQ procedure has been started (i.e. search state). Note that for all resolved nodes the HIT is always known.

When possible and to avoid bandwidth waste, LUNAR also uses the NameDB table in order to reply to DNS-PTR requests for the net.lunar domain. We remind that the goal of this *inverse resolution* procedure is to obtain the DNS name associated with a given IP address. One must note that with the traditional DNS operation, a host will send a PTR request to its DNS server even if it just resolved the IP address of the corresponding name. This additional overhead occurs because current operating systems do not implement a name cache and therefore a previously resolved *name → IP address* mapping cannot be re-used to perform the inverse resolution. In contrast, if a net.lunar name has recently been resolved into the corresponding IP address by a given node N, no PTR request is sent into the MANET if the node N wants to resolve this IP address into a name, as we use the NameDB table as a cache. Finally, to cope with the volatility of ad hoc networks, the name cache is periodically drained in order to handle address or name changes.

V. Open issues

An open issue is the case of two nodes picking identical hostnames in their FQDN. For example, two nodes respectively named *john.domain1.net* and *john.domain2.com* will both end up being identified inside the MANET as

john.net.lunar. To resolve this issue, we plan to add a mechanism to check for duplicate names at the same time when we check for duplicate IP addresses i.e., with the same RREQ message. Similar to picking another random IP address in case of a collission, LUNAR will start adding a suffix to the hostname and test again with the new name. For the previous example, the two nodes would thus end up being for example named *john.net.lunar* and *john33.net.lunar*.

Note that the new name is only used inside LUNAR i.e., at layer 2.5 and is mostly relevant to the *other* MANET nodes trying to contact the node with the new name: No attempts are made to change the host's original FQDN (*john.domain2.com*), which (a) would be a challenging implementation excercise and (b) could also be an unwanted source of confusion to the end user. In other words: we keep a mapping table between the new names and the LUNAR IP addresses, not the old (derived from FQDN) names and the old IP addresses. To allow users to distinguish between *john* and *john33* which with very high probability should have different HITs, a user could use the `/proc/net/lunar` file to distinguish the two nodes if the HIT is derived from a well-known public key identity (see [12]).

A second issue relates to the case of merging network clouds: This can lead to a MANET where some hosts have identical IP addresses[4] and/or identical LUNAR names. We (already) solve this problem by introducing stealth "host identifier tags" (HITs): any node joining the network with a colliding name or IP address will not be discovered by the LUNAR path establishment procedure as it has a different HIT. This strategy, which was proposed by [13] and which is inspired from [12], permits to maintain TCP connections although new hosts appeared in the MANET with the same IP address. However, we still need to implement a scheme to perform a large-scale IP(v4) renumbering.

VI. Conclusion

In this paper, we have shown and demonstrated how a MANET network can exhibit full Internet integration, thanks to the use of a dedicated underlay scheme which positions itself just below the IP layer. A key insight of this paper is that this underlay has to incorporate a model of the IP stack that it serves, should it wish to create a perfect fixed Internet illusion for it.

In addition to Internet integration, the underlay scheme also allows to optimize network operations with respect to the specific properties of distributed wireless ad hoc networking. This protocol fusion permits to merge the network operations of name resolution, link-layer address resolution, and network path setup in a single and efficient procedure. In particular this is done without any modifications of the existing operating systems, applications, and name resolver library. Since MANETs break several implicit assumptions (like client-server orientation) of IP networking, our underlay approach permits to rearrange basic functionalities in a MANET-friendly way. It is a first step towards a functional re-composition of IP-related protocols outside layering constraints.

[4]Note that address collision is merly an IPv4 issue, since the probability of IPv6 address collisions is very low, thanks to the use of Ethernet Unique Identifiers of 64 bits (EUI-64).

References

[1] S. Shakkottai, T. Rappaport, and P. Karlsson, "Cross-Layer Design for Wireless Networks," *IEEE Comm. Mag.*, vol. 41, no. 10, pp. 74–80, October 2003.

[2] C. Jelger and C. Tschudin, "Underlay Fusion of DNS, ARP/ND, and Path Resolution in MANETs," in *Proceedings of ADHOC'05*, May 2005, Stockholm, Sweden.

[3] P. Engelstad, D. V. Thanh, and G. Egeland, "Name Resolution in On-Demand MANETs and over External IP Networks," in *Proceedings of IEEE ICC'03*, May 2003, Anchorage, Alaska.

[4] J. Jeong, J. Park, and H. Kim, "Name Service in IPv6 Mobile Ad-hoc Network connected to the Internet," in *Proceedings of IEEE PIMRC'03*, Sept. 2003, Beijing, China.

[5] C. Perkins, E. Belding-Royer, and S. Das, "RFC 3561 - Ad hoc On-Demand Distance Vector (AODV) Routing," July 2003.

[6] D. Johnson, D. Maltz, and Y.-C. Hu, "Internet Draft - The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR), draft-ietf-manet-dsr-10.txt," July 2004.

[7] S. Thomson and T. Narten, "RFC-2462 - IPv6 Stateless Address Autoconfiguration," December 1998.

[8] T. Narten, E. Nordmark, and W. Simpson, "RFC 2461 - Neighbor Discovery for IP Version 6 (IPv6)," December 1998.

[9] C. Tschudin, R. Gold, O. Rensfeld, and O. Wibling, "LUNAR - A Lightweight Underlay Network Ad-Hoc Routing Protocol and Implementation," in *Proceedings of NEW2AN'04*, February 2004, St. Petersburg, Russia.

[10] C. Tschudin and R. Gold, "Network Pointers," in *Proceedings of First ACM Workshop on Hot Topics in Networks (HotNets-I)*, October 2002, Princeton, NJ, USA.

[11] C. Jelger and T. Noel, "Proactive Address Autoconfiguration and Prefix Continuity in IPv6 Hybrid Ad Hoc Networks," in *Proceedings of IEEE SECON'05*, September 2005, Santa Clara, CA, USA.

[12] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Internet Draft - Host Identity Protocol, draft-ietf-hip-base-03.txt," June 2005.

[13] N. Vaidya, "Duplicate Address Detection in Mobile Ad Hoc Networks," in *Proceedings of ACM Mobihoc'02*, June 2002, Lausanne, Switzerland.

A Cross-Layering and Autonomic Approach to Optimized Seamless Handover

G. A. Di Caro*, S. Giordano, *Member IEEE*†, M. Kulig†, D. Lenzarini *Member IEEE*§, A. Puiatti† and F. Schwitter†
*Istituto "Dalle Molle" di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, CH-6928 Manno, Switzerland
Email: gianni@idsia.ch
†University of Applied Sciences (SUPSI), Galleria 2, CH-6928 Manno, Switzerland
Email: {silvia.giordano,kulig,alessandro.puiatti,schwitter}@supsi.ch
§Forward Information Technologies SA, Galleria 2, CH-6928 Manno, Switzerland
Email: davide.lenzarini@forit.ch

***Abstract*— Performing global node mobility requires to support seamless vertical and horizontal handovers between different network providers and technologies. The optimization of the handover processes requires not only continuous network services, but also continuous service performance. In this paper we present a novel design approach based on autonomic components and cross-layer monitoring and control to optimize the performance of the WiOptiMo system, which provides seamless internetwork roaming by handling mobility at the application layer. We also distinguish from past work, in that we pragmatically followed the approach to rely only on existing technologies, deployed protocols and lightweight calculations, such that our system can be straightforwardly implemented as it is on most of the currently available mobile network devices. We report results from some simple real-world experiments showing the benefits of using a cross-layering approach. We also describe a first version of the new WiOptiMo based on the innovative design. Results from preliminary tests in real-world scenarios indicate the effectivness of our pragmatic approach. The system is still under development and testing, and we plan to integrate in it several autonomic components, which are presented and discussed in the paper.**[1]

I. Introduction

In forthcoming scenarios for heterogenous mobile networks users can benefit of ubiquitous coverage by roaming between different access networks (*internetwork roaming*). The challenge consists in providing seamless and continuous connectivity with the QoS required by the user applications given the different characteristics (in terms of coverage, bandwidth, cost, etc.) of the different available networks. The optimization of the decisions and procedures of internetwork roaming is a key issue for the successful deployment of pervasive and ubiquitous network services. While several solutions have been proposed in the past for the internetworking roaming problem, usually they are either too complex to be efficiently implemented in real-world applications, or they assume the access to parameters and information which are hardly available given current standard protocols and devices.

The solutions we present in this paper, integrated in the *WiOptiMo* [1], [2] system, have been designed to overcome the drawbacks of previous work. WiOptiMo is a middleware system based on a pair of Client-Server modules at the application layer. These modules interface the communication between the actual Client and Server applications hiding the mobility to them, by letting them "believe" that they run on the same machine.

While these core characteristics of the WiOptiMo approach have been left unchanged, in this paper we present several improvements of the system in terms of self-tuning and adaptation of parameters and optimization strategies, collected information, and optimization of the adopted decision procedures on the basis on this same information. The proposed modifications are the result of a novel design approach based on the use of *autonomic* [3], [4] components and *cross-layer* [5], [6] monitoring and control. We claim that this is the way the go to deal effectively with the challenges posed by the dynamic, probabilistic, and technological aspects of the multiobjective optimization problem at hand. More specifically, the proposed novel approach, which is based on the collaboration among *physical*, *network* and *application* layers to act efficiently at the network layer, is expected to boost-up system performance, as well as to show a good level of *adaptivity*, which is a key component to properly act and react in the highly dynamic scenarios of interest. Furthermore, the architectural innovation is supported by autonomic components: the system is able to *self-configure* and to *self-optimize* its internal parameters when introduced in a novel hardware/network context, and in the future it is expected to use past experience to show anticipatory behaviors and/or learn about user preferences. As opposite to what hase been proposed in the past for the internetworking roaming problem, in the *WiOptiMo* system we pragmatically followed the approach to rely only on *existing technologies*, *deployed protocols and lightweight calculations*, such that our system can be straightforwardly implemented as it is on most of the currently available mobile network devices. Clearly, our approach has the "drawback" that we have to face challenging limitations in terms of information available to take statistically sound and optimized decisions, and in terms of complexity of the implemented algorithms.

This paper has to be seen as a bridge between the original WiOptiMo system, based on traditional design, and the new

[1]This work was partially funded by the CTI Swiss programme under the KTI-7640.1 ESPP-ES Optimised always-on solution project.

version of it, based on cross-layering and autonomic components. In fact, the new system is still under development and testing. Such that in this paper, we introduce the general motivations and characteristics of the novel approach, discuss the envisaged solutions, and report some experimental results supporting our point of view. We also describe in some details the current status of the new implementation.

The main contributions of the paper are: (1) the introduction of a novel cross-layering and autonomic approach for seamless and efficient internetwork roaming, (2) a set of experimental results from real-world scenarios providing a first validation of the soundness of the approach, and (3) the description of a beta implementation of the new WiOptiMo, based on a minimalist cross-layering and autonomic design to allow the effective and portable implementation on commercial platforms.

The rest of the paper is organized as follows. Next section reports a short summary of the general characteristics of the internetwork roaming problem and of the proposed solution approaches. Section III and its subsections describe the general architecture and the specific components of the original WiOptiMo system. Section IV provides the generalities about the cross-layering and autonomic design approach. Subsection IV-A discusses the proposed solutions for cross-layer monitoring and control at the physical and application layers. Subsection IV-B describes the autonomic components of WiOptiMo, while Subsection IV-C discusses the components of the system dealing with user interaction. Section V reports results of real-world experiments aimed at showing the effectiveness of application layer active measures to infer traffic load information. Section VI describes the new implementation of the Check Activity, one of the main components of WiOptiMo, redesigned according to the cross-layering and autonomic point of view. Finally, Section VII draws some conclusions and discusses future work.

II. Definitions and General Characteristics of the Internetwork Roaming Problem

The problem of internetwork roaming is also referred to as the problem of network *handover* (or *handoff*). The switching between two different types of networks is called *vertical handover*, while *horizontal handover* refers to the case of networks of the same type. The handover can be either *soft* (or *alternative*) when it is executed for the sole purpose of *optimization* of the connection cost or QoS, or *hard* (also termed *imperative*), when it is executed due to imminent or present loss of connectivity.

The handover of the mobile terminal (MT) is *network executed* if it is done by the network connection point (e.g., as it is the case between UMTS/GSM/GPRS cells). *Mobile executed handover* is the case in which the handover decision is autonomously taken by the MT. This is the modality prescribed by the currently deployed 802.11 standard for WLANs and this is the main focus of our work. For mobile executed handovers the strategies for horizontal handovers between WLANs, and vertical handovers from WLAN to WWAN and vice versa, have slightly different characteristics due to the existing differences in the information available to the MT at decision time and in terms of coverage, bandwidth, cost, and signal strength between the two different types of networks.

Building-up on previous work for handover in GSM networks, the majority of the approaches for horizontal handover in WLANs reason on the "quality" of a connection making use of simple *threshold-based schemes* usually including *hysteresis margins* [7] and *dwell timers* to enhance overall robustness (e.g., see [8]). The quality of a connection can be expressed by means of any desired combination of metrics related to PHY measures of the quality of the received signal and/or to MAC/TCP/APP measures of the available bandwidth. It is common practice to restrict the use to the signal strength and/or the signal-to-noise ratio of the received signal, and to measure the amount of packet losses. A handover is executed only if the new AP seems to provide better connection quality over some stability/dwell period, avoiding a "ping-pong" effect when the MT crosses the overlapping edges of two cells and the signal from the connected AP usually shows significant fluctuations. Considering that each AP switch involves time-consuming *authentication* and *reassociation* procedures (e.g., see [9]), it is important to avoid unnecessary handovers by carefully assigning the values of all the used parameters and thresholds.

Also in the case of vertical handover from WLAN to WWAN and vice versa, a threshold-based scheme (sometimes combined with additional adaptive or fuzzy mechanisms) has been adopted in the majority of the studies(e.g, [10], [11], [12], [13], [14]). Using dual-mode or multiple NICs, it is possible to monitor alternative connection points while using the current network connection (even if this has a negative impact of the on-board energy consumption). Due to the bandwidth and coverage differences existing between WLANs and WWANs, as rule of thumb most of the decision schemes tend to always favor WLAN to WWAN handovers (*upward link* handovers) in case of high speed, while tend to favor WWAN to WLAN handovers (*backward link* handovers) for low speeds. Often, load/bandwidth information is not even took into account, relying on the fact that WLAN's bandwidth outperforms WWAN's bandwidth even under quite high loads such that it is always preferable to switch to a WLAN (see [12] for an argument against this strategy). This approach is also motivated by the intrinsic difficulty for an MT to derive sound estimates of the *available bandwidth* (see also [15], [16] for related work in wired and wireless environments). On the other hand, for the connection points it is relatively easy to measure it, such that they could provide this information to help the MT to take its decisions (realizing a mobile executed but *network assisted handovers*). Unfortunately, given the current status and practical implementations of the IEEE 802.11 standards, in current WLANs MTs cannot rely on *any* feedback from the APs. Indeed, this fact puts strong limitations on the actual optimization of mobile executed handovers (the activities of the IEEE 802.11k working group are precisely addressed to remove these limitations in the deployed standards). However, the work is still in progress and it is still unclear how soon and

if the new specifications will be formally adopted (see [17] for a discussion about problems related to disclosing user/network information in both network and mobile assisted handovers).

Using the same terminology adopted in [13], the whole handover process can be practically decomposed in to three functional blocks:

- *Handover Initiation*
- *Network Selection*
- *Handover Execution*

Handover Initiation consists of the *proactive monitoring* of the current connection and/or of possible alternative connections in order to: (i) effectively *anticipate* or explicitly deal with imperative handovers, or (ii) trigger alternative handovers in order to *optimize* costs and performance. In our case, the CNAPT *Search* and *Check Activities* (see Section III-B) both participate to the Handover Initiation process, which is however mostly focused on the treatment of imperative handovers. Network Selection comprises the procedures to select the new connection point according to *decision metrics* like quality of the signal, cost, bandwidth. etc.. Information about these metrics can be gathered either proactively and/or reactively according to the proposed scheme and to the limitations imposed by the used protocols and technology. In our case, Network Selection is supported by the results provided by the Search Activity. Handover Execution stands for the set of procedures to be carried out for the authentication and reassociation of the MT. This pertains to the WiOptiMo CNAPT/SNAPT switching procedure (see Section III-A).

Assessing the goodness of the current and alternative network connections requires, for both Handover Initiation and Network Selection, the proactive and/or reactive passive gathering of PHY data concerning the behavior of signal quality and/or MAC/TCP/APP layer data about the effective bandwidth and latency associated to the wireless link (e.g., [10], [18]). More complex approaches consider also mobility/location information [19], or learning user preferences and behavior [20]. The use of active monitoring based on probing packets to estimate load conditions has also received some attention [16].

III. The WiOptiMo System

WiOptiMo [1], [2], [21] is a solution for seamless handover among heterogeneous networks/providers. That is, it transparently provides persistent connectivity to users moving across different wired and wireless networks. WiOptiMo detects the available network access points and provides, in automatic or semi-automatic/assisted way, the best Internet connection in terms of estimated QoS (e.g., bandwidth, reliability, and security) and/or cost effectiveness among all the available connections at a certain time and location. The optimized handover is executed without interrupting active network applications or sessions and avoiding or minimizing user intervention. Furthermore, if the current connection becomes no longer available and if no other connections can be established (e.g., inside an uncovered area), the system hibernates the applications to perform re-establishment when the current or a new connection becomes available again (obviously, if the reestablishment exceeds the application timeout, the application may detect a network problem).

In the following subsections we briefly describe the main characteristics of the original WiOptiMo system. From Section IV onward, we discuss the rationale behind the design of the new version of WiOptiMo, and we present the already implemented modifications as well as the general characteristics of the modifications that are still under implementation and testing.

A. WiOptiMo Application Layer Solution

The WiOptiMo system does not require any modification of the layers of the OSI protocol stack and it does not introduce any additional sub-layer. The seamless handover is obtained by a pair of applications (OSI Layer 7), the *CNAPT* (Client Network Address and Port Translator) and the *SNAPT* (Server Network Address and Port Translator), which deceive the communicating Client and the Server applications letting them believe that they are running on the same device, or on different devices belonging to the same network. The Client and the Server applications do not realize that they are communicating via the Internet. The CNAPT and the SNAPT collectively act as a *middleware* and interface the communication between the Client and the Server applications hiding the mobility to them. The CNAPT is an application that can be installed in the same device as the Client application or in a different device in the same mobile network (e.g., in the case of a team of consultants or auditors that require mobility while working together, the CNAPT can be installed in only one of the mobile devices of the mobile network and the whole team can share the seamless handover provided by it). The SNAPT is an application that can be installed in the same device as the Server application or in a different device of the same network or in any Internet server (e.g., in a corporate front-end server, in the home PC, or in any Internet node or router). Thanks to this flexibility, the mobility of multiple users can be handled either using a star topology, with central servers with large computational capabilities and large bandwidth and managed by telecommunication companies or ISP, or using a distributed topology, in which every user manages its mobility by installing the SNAPT on the accessible nodes (e.g., in the home PC if directly connected to the Internet), saving in terms of transmission costs. With the distributed topology, WiOptiMo can provide a sort of ”democratic” seamless handover.

B. WiOptiMo Handover Initiation and Network Selection: Search *and* Check *Activities*

The CNAPT application acts as an application relay system, and also activates a decision task in order to provide persistent and optimized Internet connectivity. The decision task consists of two main activities: the *Search Activity*, for soft handovers, which proactively searches for new network providers and connectivity, and the *Check Activity*, for hard handovers, which continuously monitors reliability and performance of the current connection. Moreover, at any time, the user can

manually ask the decision task to switch to another available network connection. This can be useful when the user wants to use a specific network that would be not selected otherwise.

1) Search Activity: The Search Activity periodically searches for other available network connections. In the forthcoming new version of the system, this will be done without disturbing the current connection, even in the case that both the current and the checked connection are WLAN (the new behavior builds on the results reported in [22]). The handover to a new network can be triggered in either manual or automatic mode. In *manual mode*, when the Search Activity finds at least one network that could provide an Internet connection better than the current one (based on the parameters input by the user), it asks the user whether she/he wants to switch. In *automatic mode*, the Search Activity autonomously decides whether to execute or not the handover (again, the decision is made on the basis of a set of parameters assigned by the user). In some cases, in automatic mode, the Search Activity might still require some minimal user interaction to the complete the network association procedures. In all cases, from the running application point of view, the Search Activity avoids any interruption of the service during the handover. After the handover has been performed, the user can choose to keep the old Internet connection, otherwise it will be closed in order to reduce power consumption.

2) Check Activity: Following a periodic activation scheme, the Check Activity verifies reliability and performance of the current connection. In the current implementation the check interval is a parameter set to one second. If the reliability or performance index go below some specified critical thresholds (possibly set by the user), or the current network connection is experiencing an interruption, the Check Activity tries to switch to a new network provider, or tries to set up a new Internet connection from the same provider if the signal comes back (i.e., the disconnection was only a temporary problem, like when crossing a small uncovered area during a UMTS connection). If the Check Activity does not find any available network providers, it notifies the user that the current connection will be no longer available and changes its operational mode to *Tunneling mode*, which consists of a continuous search for an available network connection. When it eventually finds at least one available connection point, it automatically establishes the connection and comes back to its normal operational mode. During the switch, the Check Activity avoids any interruption of the service. After the switch has been performed, the user can still choose to close the old Internet connection, if it is still alive, in order to reduce power consumption.

IV. Cross-Layering and Autonomic Design

Given the complexity and the multiple dynamic aspects of internetwork roaming, a design based on a *cross-layering* architecture [5] interleaved with *autonomic* components [3] is almost an unescapable choice to achieve adaptive behavior and performance optimization. As it is also witnessed by other approaches to handover optimization (e.g., [10], [23], [6]), cross-layer data are necessary to collect the information necessary to deal with the intrinsic variability and uncertainty associated to the physical measures of interest like the Received Signal Strength (RSS) and the available bandwidth. On the other hand, autonomic components are based on the proactive monitoring of the system's performance and behavior which, in turn, results in the self-optimization of internal thresholds and in the continual adaptation of the delivered Quality of Service (QoS) to the ever changing external conditions. Adopting this approach, the user can experience seamless connectivity in a fully transparent way, letting the system to adapt to his/her needs and to struggle to provide the required QoS.

The general architecture of the new WiOptiMo system consists of three main functional components: (i) *cross-layer monitoring*, that performs active and passive monitoring activities and data collection at the physical, network, and application layers, (ii) *self-optimization and learning*, carried out by an *optimization module* which makes use of a repository of past experiences to adapt internal parameters and derive statistical measures of trend, (iii) *interaction with the user*, intended to get user's feedback for the selected handovers and to allow the user to input its preferences' profile through an intuitive interface.

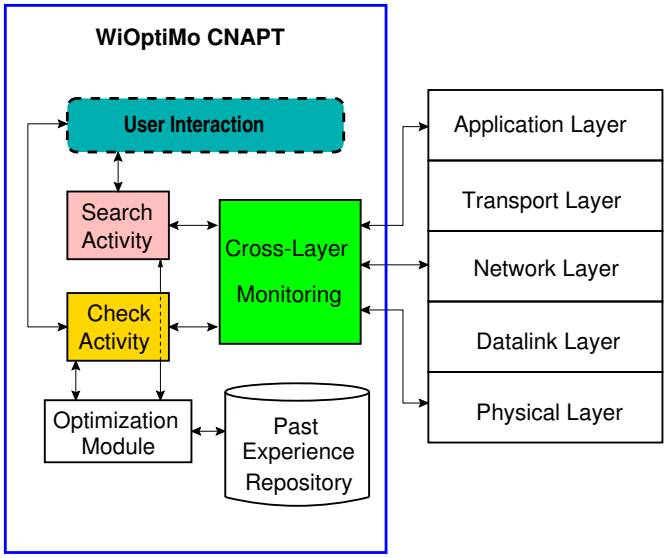


Fig. 1. Functional components in the WiOptiMo CNAPT module

Both the original Check Activity and Search Activity components have been re-designed according to this combination of cross-layering and autonomic approach. In the following we provide a general discussion on cross-layering and autonomic issues in WiOptiMo, as well as a description of their implementation in the new version of WiOptiMo. However, since the new release of the system is still undergoing full development and testing, we keep the discussion at a rather general level for those aspects that have not been fully implemented and carefully tested.

A. Cross-Layer Monitoring of Network Connections

Simple PHY monitoring, while necessary to understand the physical status of the connection, is unable to offer information on the connection traffic load. On the other hand, we could not take into account LINK monitoring of variables related to frame count, since apparently some NICs do not allow to read these values using standard APIs. Therefore, in our novel design, we integrated the application layer approach of WiOptiMo system with the access to information and protocols belonging to other layers, and in particular to the physical layer. From one side, this solution introduces interdependencies of the WiOptiMo system with other layers, which create additional complexity at design and implementation time. On the other side, we observed in practice that this solution can really allow to optimize system performance without having a negative impact on the overall efficiency of the system.

1) Physical Layer Monitoring: ¿From our experiments, we derived that physical monitoring is significant and reliable only for few parameters, namely the RSS and FCS, and we decided to use the RSS, as the FCS does not offer more information than the RSS [21], [24]. In the original WiOptiMo system, we decided to perform periodic (every second) sampling of the RSS. The RSS value is used raw, without further smoothing. The use of raw values and such a low sampling rate was justified by the need of keeping as low as possible the computational load on the mobile device. However, due to the high and frequent fluctuations of the raw RSS, we realized that we need richer and better information to derive proper conclusions on the evolution of the RSS. At this aim, in the new version the system will read RSS information for each received frame. In this way, it will have sufficient data to be able to robustly smooth the sampled values through simple weighted moving averages, and at the same time calculate a simple trend indicator to be used in cross-validation with the moving average (both the chosen statistical indicators can be efficiently implemented using elementary integer calculations).

2) Application Layer Monitoring: WiOptiMo conveys all the traffic exchanged between the client and the server applications on the CNAPT/SNAPT connection, and the CNAPT knows at each time whether there is or not some traffic. Therefore the system can easily monitor the connection at this layer, perform further check on the quality of the connection, and derive useful information. While it is not feasible to interact with the information contained in all the exchanged data packets, as it would be too expensive, we found appropriate to carry out active measurements at this layer by injecting few control packets in the connection and observe their behavior. More specifically, we send *ping messages* (ICMP packets) to the access point, for the purpose of:

- Estimate the connection in terms of available throughput,on the basis of the experienced Round-Trip-Time (RTT). This gives some indication on:
  1) the *effective status* of the connection (if it is active or not), as the status detected at the physical layer could be not correct, as discussed in Section VI
  2) the *effective load* of the connection, which would not be possible to infer from the measures available at physical layer. We are aware that ICMP traffic is controll traffic. However, as we are not trying to quantitatively measure the throughput, but we simply want to have a dynamic estimation of the WLAN charge, this is sufficient for our purpose.
- Avoid to improperly react to temporary fluctuations of physical parameters. While a negative indication from physical monitoring (e.g., a very low RSS level) corresponds to a high probability that the physical connection (and consequently the CNAPT/SNAPT connection) is not anymore valid, this can be due just to some temporary cause. On the other hand, if a short train of ping messages experience a timeout, this evidence can provide a more robust confirmation of the fact that a link is down.

B. Autonomic Design

In previous work [21], [24] we conducted extensive experimental tests in real-world scenarios to gain insights on the possible metrics and strategies that can be adopted to optimize the procedures of internetwork roaming. One of the results that has emerged from our experiments is that *the goodness of a metric or of a parameter value, strongly depends on the wireless scenario at hand*. For instance, the behavior of the RSS can be a good or bad predictor of the quality of the connection depending on the considered scenario and on the specific hardware used. Therefore, this tells us that an optimization mechanism based on static parameters and strategies is not apt to deal satisfactorily with the large variety of wireless scenarios of practical interest. Our approach to solve this issue is to *empower the cross-layering architecture with an autonomic design*: our system is able to *adapt* and *self-optimize/tune* its internal parameters and performance according to an understanding of both the current context (wireless scenario inclusive of the specifically used equipment) and the user preferences and mobility patterns. In order to show a truly adaptive behavior, and in some extent predict and anticipate changing in the environment and in the user activities/mobility, it is necessary to *learn from past experience*. At this aim, the new WiOptiMo will include a repository of the most significant information (trends, failures, trajectories, user choices, etc.) about past experience. For instance, if an on-board GPS is available, the system can decide to store and learn maps identifying good coverage areas together with the characteristics of the network access that can provide the coverage. This might result quite useful (and relatively easy to learn) in the case of users constantly travel along the same routes, as it is the case for people daily commuting between their homes and work places.

The WiOptiMo system is intended to run unchanged on the majority of the portable devices in commerce. This is reflected in the fact that the system can self-detect the characteristics of the on-board hardware and take the appropriate action flows (see Section VI). This is a fundamental *self-configuration* feature already present in the system, that, together with

the discussed properties of self-optimization and and self-tuning, support even more the view of WiOptiMo modules as autonomic modules.

1) A concrete example: Self-tuning of the RSS threshold: The default RSS threshold value that we used in the original WiOptiMo system to check connection reliability (in both the Check and Search activities), does not always reflect the characteristics of the current context in terms of signal characteristics at the specific location, and hardware and software configuration. In order to achieve adaptive context-dependent tuning, we designed a simple adaptive component for the self-tuning of the RSS threshold according to the current context. We continuously check if the system, when it looses the connection (i.e., an IP address is not anymore available), has reached an RSS value lower than that of the currently stored RSS threshold value. If so, we assume that this new lower value is the inferior limit for successful communications at the current location, given the hardware and software configuration. Therefore, we take this new value to adapt the value of the variable containing the RSS threshold, and we start using it. This behavior can be started either by the system, under reception of an event of loss connection, or by the users, for example when she/he installs a new hardware. The current implementation, which can be only started manually, requires that the user moves around for a while, in order to gather the required information. However, in the final release of the next version we intend to fully automate the procedure.

C. User Interaction

An important part in the definition of the context that the optimization strategy has to take into account, is played by the user itself, with its current preferences. In many cases, in practice it is not possible to assume that the system can infer/learn user's preferences in relationship to its needs at the current time and location. In order to provide a satisfactory service some user interaction must be assumed. For instance, if a specific download is very urgent, bandwidth might be more important than cost, but this cannot be known in advance to the system without additional information from the user, who has to input is current QoS requirements in its *user profile*. Moreover, some additional information concerning for instance the "typical" current speed of the user (e.g., vehicular, pedestrian, etc.) might be of great help to the system, even if this information might be inferred from lower layer data. At this aim, we are realizing an intuitive and user-friendly user interface, to facilitate the input and the updating of profiling information. We are also considering mechanisms to trigger requests of user-assistance in case of very ambiguous situations.

V. Experimental Results for Measures of Load Estimation at the Application Layer

Experimental results concerning work in the domain of vertical handover optimization are mainly restricted to simulations (a notable exception is reported in [25]). An important negative consequence of this consists in the fact that it is really hard to assess the validity of the proposed simulation-based approaches in the perspective of the implementation in real-world networks [26]. On the other hand, as already pointed out, our choice is to face the real-world challenges. In this perspective, and we the aim of gathering experience for a first validation for the use of cross-layering monitoring, we realized a set of real-world experiments focused on the use of ping messages (see IV-A.2) for the establishment of the effective status of the connection, and for channel load estimation.

The soundness and performance of the (original) WiOptiMo system in the case of using only simple physical monitoring was assessed in previous work [1]. In [21] we reported extensive results concerning the effectiveness of a number of different metrics at the physical and MAC layers. Here we present a set of experiments aimed at assessing the utility and the usability of integrating physical layer controls with application layer controls. The results of these experiments gave us a further motivation to proceed along the way of integrating cross-layer monitoring and control in the WiOptiMo system, as it is discussed in the next section.

All the experiments were conducted in favorable conditions of strong and stable RSS, to see if, given good physical conditions for the connection, we could observe different indications at the application layer in relationship to variations in the wireless scenario. More specifically, we report the results concerning the use of ping messages to derive load estimates at the application layer as a function of variations in the load offered to the wireless environment.

Ping messages are proactively sent to the access point in order to evaluate the connection status in terms of: (i) *throughput*, on the basis of the observed RTTs, and (ii) *channel congestion* according to the number of experienced timeouts. To understand whether or not the RTT of ping messages can be used as a robust indicator of channel congestion, we performed traffic-intensive experiments with an increasing load offered to the WLAN. We considered the following four scenarios: (1) an increasing number of laptops (from 1 to 10) continuously download data from one PC connected via LAN to the AP, while another laptop sends the ping messages to the AP; (2) an increasing number of laptops (from 1 to 7) continuously download data from each other passing through the AP, while another laptop sends ping messages to the AP (in this way the network load was doubled with respect to the previous case); (3) an increasing number of laptops (from 1 to 10) continuously download data from the Internet through the AP, while another laptop sends the ping messages to the AP; (4) 11 laptops with an increasing number of downloading processes (from 1 to 5 on each laptop) from the Internet through the AP, while another laptop sends the ping messages to the AP. In each scenario a continuous train of 1000 ping messages (size of 32 bytes) was generated during the experiment. The size of each single download was of 1.5 Gbytes. The WLAN is based on 802.11g devices.

As shown in Figures 2 and 3, in the first two scenarios the average RTT and the number of experienced timeouts

increased each time that a new laptop joined the network with a new download. Moreover, the number of timeout rose up very fast when the number of laptops in download grew from 7 to 10 in the first scenario, and from 3 to 7 in the second scenario.

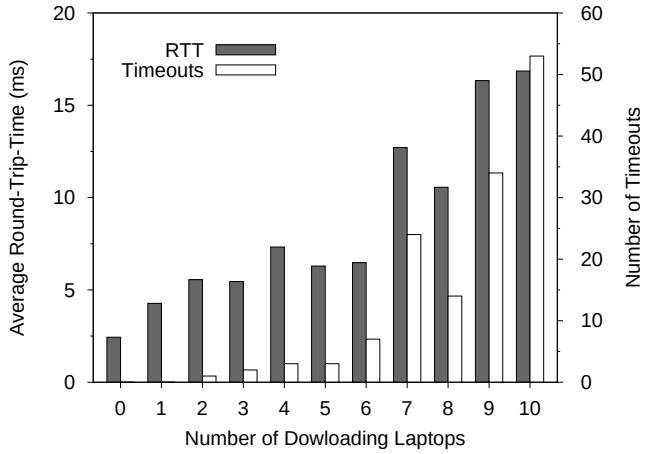


Fig. 2. Experimental results for the first load estimation scenario. An increasing number of laptops download data from one PC connected via LAN to the AP while another laptop sends ping messages to the AP.

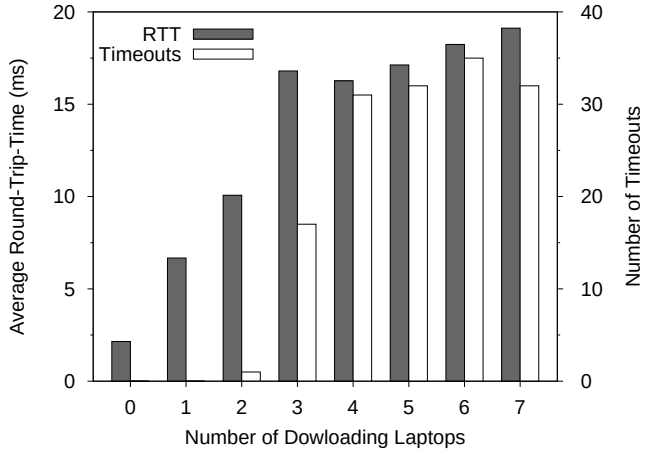


Fig. 3. Experimental results for the second load estimation scenario. An increasing number of laptops download data from each other passing through the AP while another laptop sends ping messages to the AP.

In these scenarios, the RSS was always above the critical threshold. Therefore, no handover indications could have been derived from this physical measure. However, from Figures 2 and 3, it is clear that the throughput was significantly decreasing, and the channel congestion increasing, with the increase of the active users in the WLAN. Therefore, the combined analysis of the RSS and RTT values seems to be necessary to become aware of a situation of pure traffic congestion.

On the contrary, in the third scenario (Figure 4) the both the average RTT and the number of timeouts remain low almost independently from the number of laptops in download. This was probably due to the available bandwidth on the Internet side, that was lower than that available in the WLAN. In the last scenario we overloaded the WLAN with a large number of downloads (11 up to 53) from the Internet. As shown in Figure 5, the average RTT remained more or less the same but the number of timeouts rose up very fast reaching more than 10% of the total number of ping messages sent in the worst case. Therefore, in these last two scenarios, the advantage

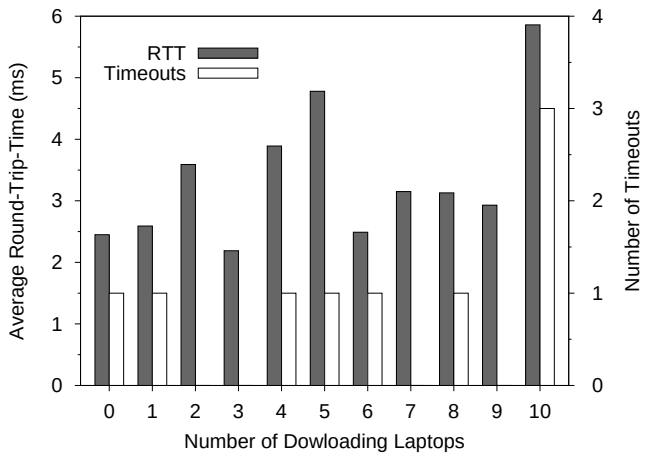


Fig. 4. Experimental results for the third load estimation scenario. An increasing number of laptops download data from the Internet through the AP while another laptop sends ping messages to the AP.

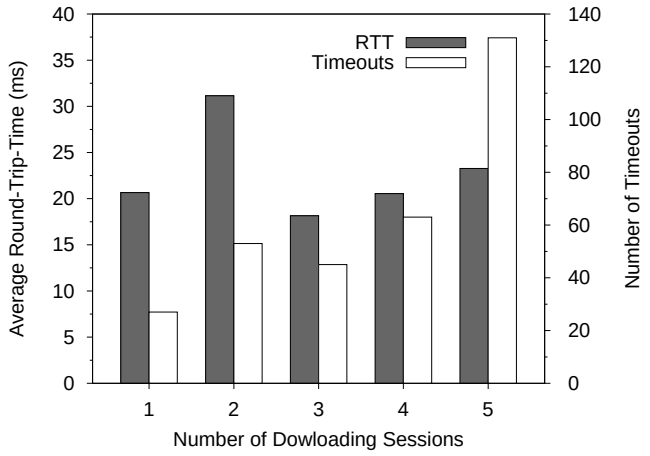


Fig. 5. Experimental results for the fourth load estimation scenario. 11 laptops with an increasing number of sessions download data from the Internet through the AP while another laptop sends ping messages to the AP.

of a cross-layering approach have been less evident, but still useful to estimate the quality of the connection throughput.

VI. The New Check Activity

The experimental results discussed in the previous section seem to indicate that the information obtainable at the application layer can fruitfully complement the information from the physical layer to derive more robust estimations of the status of a connection link. As shown by the previous experiments, we can have discordant indications between physical and

application layer monitoring of the connection, such that they should be used in cross-validation. This is one of the main reasons behind our choice to adopt a cross-layer approach and modify the Search and Check Activity according to the design ideas presented in IV. Here we present the basic structure of the new Check Activity, which includes both *passive monitoring at the physical layer* and *active monitoring at the network/application layer*. In the new Check Activity, the use of ping packets, even if not fully exploited, resulted very useful, as discussed below. The new Check Activity repeatedly interacts with the user, and tries to learn from past experience, to adapt and self-tune internal parameters and derive trend measures (for the RSS). However, we are still testing and improving these mechanisms, such that they are not further discussed in the following.

The flowchart of the actions of the new Check Activity based on cross-layering design ideas is reported in Figure 6. The flowchart shows only the stable core of the function, omitting the parts still under developments.

The Check Activity first checks the presence of the IP address for the current connection (*Network Layer check*). If the IP address is present (i.e., from the operating system point of view, the communication device miniport is connected), it verifies that the current connection is a wired connection, (e.g., Ethernet LAN, ADSL, Token ring, FDDI). This phase is called *First Physical check*. If this is the case, no additional checks are needed, as the wired connections are considered as always good and reliable connections. Otherwise, if the connection is a wireless one and it is in use that is, there is at least one application or service that needs to access the Internet (*First Application Layer check*), the Check Activity performs a number of additional actions which depend on the type of devices used to make the wireless connections. These devices are grouped in three categories:

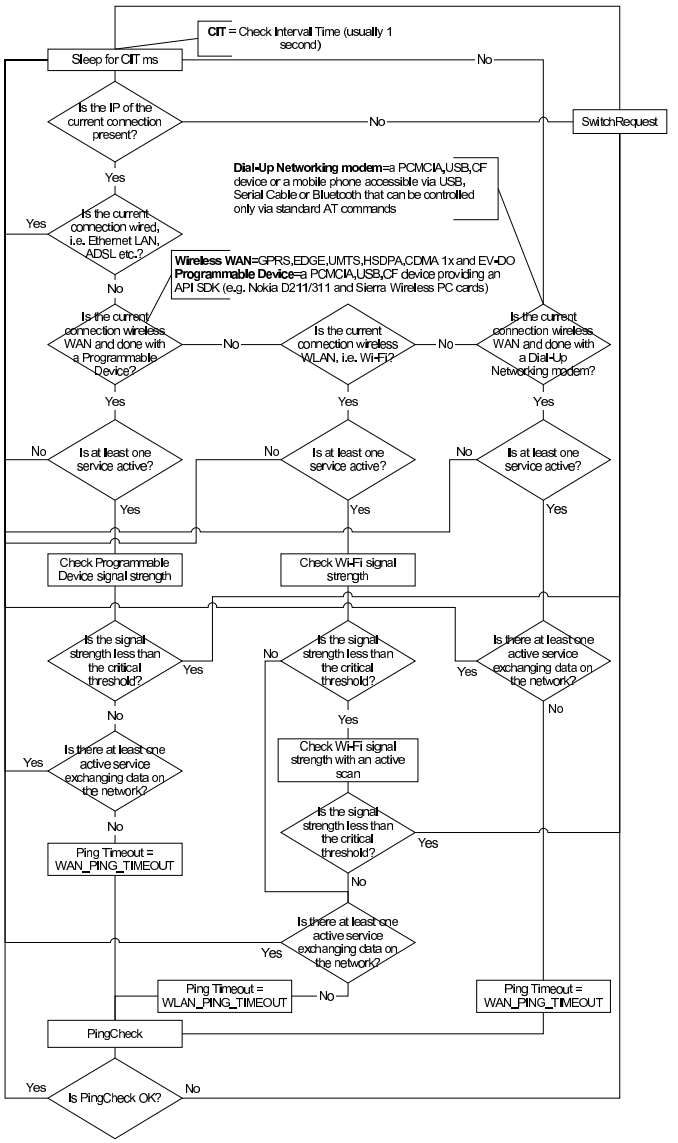


Fig. 6. Flowchart of the Check Activity function

1) Wireless WAN Programmable devices: GPRS, EDGE, UMTS, HSDPA, CDMA 1x or EV-DO PCMCIA, and USB or CF devices providing an API SDK (e.g., Nokia D211/311 and Sierra Wireless PC cards), that gives the possibility to establish/destroy the wireless connection and control all its parameters;
2) Wi-Fi devices;
3) Wireless WAN Dial-Up Networking (DUN) modems: GPRS, EDGE, UMTS, HSDPA, CDMA 1x or EV-DO PCMCIA, USB, CF devices or mobile phones accessible via USB, Serial Cable or Bluetooth that can be controlled only via standard AT commands.

In general, for the third category, once the DUN connection has been established, it is not anymore possible to access the connection parameters (e.g., the signal strength), as the COM port used for the interaction is held by the operating system to provide the WAN connection. [2] Therefore, the Check Activity cannot perform any further physical check. On the contrary, for the Wireless WAN Programmable devices and for the Wi-Fi devices, a check on the signal strength is performed (*Second Physical Layer check*). If the RSS is greater than a *critical RSS threshold*, or if a Wireless WAN Dial-Up Networking modem is used, the Check Activity controls the network activity for the active applications or services (*Second Application Layer check*). If an exchange of data is going

[2] As already discussed, WiOptiMo is intended to run unchanged on the majority of the portable devices in commerce, thus it has a modular architecture, with the core module being platform-independent (JAVA). The other modules are platform-dependent. Considered that as a matter of fact, the majority of portable devices rely on the use of some version of Windows as operating system, the other modules has been developed in Windows environment, assuming the Windows' NDIS driver for the communication with the NIC. Therefore, in the following all the references to the operating system and related issues have to be thought as referred to Windows.

on, the current connection is considered *good*. Otherwise, a *ping check* is started (*Third Application Layer check*). The ping check sends three ICMP ECHO REQUEST packets to a reachable host (or if it is not possible, it establishes three TCP connections with a reachable host), and starts a timeout timer. If all the three attempts are not echoed within the requested timeout, the connection is considered lost and a handover is requested. Actually, since it is expensive in terms of time and cost (e.g., in case of GPRS connections), the ping check is not performed at every single run of the Check Activity, but with a frequency modulated by the number of timeouts observed during previous runs. This simple but rather effective strategy is similar to that adopted in [27], where the authors show some empirical evidence that the fall of a connection can be robustly assessed after three consecutive frame losses (without considering any additional metric).

We performed some tests with this new implementation and observed that the new design is very beneficial for:

1) DUN connections. Cross-layer monitoring allows to bypass the impossibility of physical monitoring, as explained above, and is the only way to check if an IP connection is still alive or not.
2) WLAN connections. If the RSS suddenly drops, some cards indicate $-200dBm$, while some other cards keep on reporting the same RSS value as before the drop or even other values. For this reason, the reported RSS value is not always a reliable indicator of good signal. On the other hand, cross-layer monitoring provides additional information on the status of the connection, and allows to understand in reliable way if it is still alive or not.

According to these facts and experimental evidences, we are currently studying a lightweight and robust solution to use ping packets for deriving load estimates.

VII. Conclusions

In line with the ubiquitous computing vision, mobile devices are becoming part of our daily life. In the majority of the real-world scenarios, due to topology, location, and time constraints, the only way to provide persistent communications and satisfactory QoS performance, is to switch from one provider to another, always trying to stay connected using the best possible connection in terms of bandwidth, cost, and coverage. However, due to the lack of systems that can really guarantee optimized seamless handovers, this revolution has not yet become real. Several solutions have been proposed in literature, but they are hardly implementable and usable in real systems. In this paper we introduced innovative design solutions to boost-up performance and flexibility of WiOptiMo, our system for internetwork roaming.

Up to our knowledge, WiOptiMo is the first solution able to work in general real-world scenarios in terms of used network cards, providers, applications, etc. We presented an innovative cross-layering and autonomic design for the forthcoming next release of WiOptiMo, and we performed some real-world experiments to support the choice of the novel design. Considering several scenarios, we found that cross-layering monitoring and control can be very beneficial, as it can allow to discover situations requiring handovers that otherwise would go unnoticed. We also point out that we made not specific assumptions, and that we are working for a system that can work in any context.

Given the preliminary encouraging results, which validated our approach, we are designing new components for quantitative analysis that will enhance the autonomy of the system and the interaction with the user.

In our current work, we are including statistical evaluations for the results, the use of different type of traffic in the wireless network (to ensure the generality of our solution with respect to the traffic generated) and the monitoring of other parameters to further improve the response of the system. Furthermore, we are also investigating the use of techniques for efficiently smoothing the noisy RSS and for the calculation of trend measures to be used in cross-validation with the RSS to add robustness to Handover Initiation decisions. Moreover, we will explore the use of learning techniques based on the use of past experience to improve the overall autonomy of the system. This work is expect to further improve the already very satisfactory performance and flexibility of the system.

References

[1] S. Giordano, D. Lenzarini, A. Puiatti, and S. Vanini, "WiSwitch: seamleass handover between multi-provider networks," in *Proceedings of the 2nd Annual Conference on Wireless On demand Network Systems and Services (WONS)*, 2005.

[2] S. Giordano, D. Lenzarini, M. Schiavoni, and S. Vanin, "Virtual web channel: Flow aggregation for enhanced ubiquitous web access," in *Proceedings of IEEE WirelessCom*, 2005.

[3] J. Kephart and D. Chess, "The vision of autonomic computing," *IEEE Computer magazine*, pp. 41–50, January 2003.

[4] D. F. Bantz, C. Bisdikian, D. Challener, J. P. Karidis, S. Mastrianni, A. Mohindra, D. G. Shea, and M. Vanover, "Autonomic personal computing," *IBM Systems Journal*, vol. 42, no. 1, pp. 165–176, 2003.

[5] M. Conti, S. Giordano, G. Maselli, and G. Turi, "Cross-layering in mobile ad hoc network design," *IEEE Computer*, February 2004.

[6] S. Shakkottai, T. Rappaport, and P. Karlsson, "Cross-layer design for wireless networks," in *Proceedings of IEEE Communications*, 2003.

[7] N. Tripathi, J. Reed, and H. Vanlandingham, "Handoff in cellular systems," *IEEE Personal Communications Magazine*, December 1998.

[8] M.-H. Ye, Y. Liu, and H. Zhang, "The mobile IP handoff between hybrid networks," in *Proceedings of the 12th Annual IEEE Int. Symposium on Personal Indoor and Mobile Radio Communications*, 2002.

[9] S. Pack, H. Jung, T. Kwon, and Y. Choi, "SNC: A selective neighbor caching scheme for fast handoff in IEEE 802.11 wireless networks," *ACM Mobile Computing and Communications Review*, vol. June, 2005.

[10] C. Guo, Z. Guo, Q. Zhang, and W. Zhu, "A seamless and proactive end-to-end mobility solution for roaming across heterogeneous wireless networks," *Journal of Selected Areas in Communications* Special Issue on Advanced Mobility Management And QoS Protocols for Next-generation Wireless Internet, vol. 22, no. 5, June 2004.

[11] R. Inayat, R. Aibara, and K. Nishimura, "Providing seamless communications through heterogeneous wireless IP networks," in *Proceedings of International Conference on Wireless Networks*, 2004.

[12] M. Ylianttila, M. Pande, J. Mäkelä, and P. Mähönen, "Optimization scheme for mobile users performing vertical handoffs between IEEE 802.11 and GPRS/EDGE networks," in *Proceedings of Global Telecommunications Conference 2001*, 2001.

[13] K. Murray and D. Pesch, "State of the art: Admission control and mobility management in heterogeneous wireless networks," Deliverable 3.1 D1.1 of the *M-Zones* Project, May 2003.

[14] A. Majlesi and B. Khalaj, “An adaptive fuzzy logic based handoff algorithm for hybrid networks,” in *Proceedings of the International Conference on Signal Processing*, August 2002.
[15] N. Hu and P. Steenkiste, “Evaluation and characterization of available bandwidth probing techniques,” *IEEE Journal on Selected Areas on Communication*s Special Issue on Internet and WWW Measurement, Mapping, and Modeling, vol. 21, no. 6, pp. 879–894, 2003.
[16] A. Johnsson, “Bandwidth measurements in wired and wireless networks,” Ph.D. dissertation, Mälardalen Research and Technology Centre (MRTC), Mälardalen University, Mälardalen, Sweden, April 2005.
[17] W. Zhang, J. Jaehnert, and K. Dolzer, “Design and evaluation of a handover decision strategy for 4th generation mobile networks,” in *Proceedings of the 57th Semiannual Vehicular Tech. Conf. (VTC)*, 2003.
[18] B. D. Noble and M. Satyanarayanan, “Experience with adaptive mobile applications in Odyssey,” *Mobile Networks and Appls.*, vol. 4, 1999.
[19] A. Saleh, “A location-aided decision algorithm for handoff across heterogeneous wireless overlay networks,” Master’s thesis, Virginia Polytechnic Institute and State University, 2004.
[20] G. Lee, P. Faratin, S. Bauer, and J. Wroclawski, “A user-guided cognitive agent for network service selection in pervasive computing environments,” in *Proceedings of 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2004.
[21] S. Giordano, M. Kulig, D. Lenzarini, A. Puiatti, F. Schwitter, and S. Vanin, “WiOptiMo: Optimised seamless handover,” in *Proceedings of IEEE WPMC*, 2005.
[22] R. Chandra, P. Bahl, and V. Bahl, “MultiNet: connecting to multiple IEEE 802.11 networks using a single woreless card,” in *Proceedings of IEEE INFOCOM*, 2004.
[23] K. Byoung-Jo, “A network service for providing wireless channel information for adaptive mobile applications: Proposal,” in *Proceedings of the ICC01*, vol. 5, 2001, pp. 1345–1351.
[24] M. Kulig and F. Schwitter, “Monitoraggio dei parametri per ottimizzare una soluzione di always-on,” in *Semester Project Report - SUPSI*, 2005.
[25] P. Vidales, G. Mapp, F. Stajano, and J. Crowcroft, “A practical approach for 4G systems: Deployment of overlay networks,” in *Proceedings of Tridentcom*, 2005.
[26] D. Kotz, C. Newport, R. Gray, Y. Yuan, and C. Elliot, “Experimental evaluation of wireless simulation assumptions,” 2004.
[27] H. Velayos and G. Karlsson, “Techniques to reduce ieee 802.11b mac layer handover time,” Kungliga Tekniska Hgskolan (KTH), Stockholm, Sweden, Tech. Rep. TRITA-IMIT-LCN R 03:02, April 2003.

Scheduling in 802.11e: Open-Loop or Closed-Loop?

Paolo Larcheri, Renato Lo Cigno
Dipartimento di Informatica e Telecomunicazioni – Università di Trento
Via Sommarive, 14 – 38050 Povo, Trento, Italy
e-mail:paolo.larcheri@gmail.com,locigno@dit.unitn.it

Abstract— **Scheduling in 802.11e networks is managed by the Hybrid Coordination Function (HCF), located within the access point. HCF has access both to traffic descriptions (TSPEC) and to feedback information sent in every frame by stations. In this paper we discuss the use of open-loop or closed-loop scheduling; the first one is based only on the TSPECs, while the second one relays also on the feedback information from stations and builds upon classical control theory design.**

We discuss how closed-loop scheduling can be used to manage both the QoS guaranteed traffic and the best-effort traffic with a non-marginal performance improvement compared to open-loop scheduling algorithms. We propose a simple max-min fair scheduling algorithm based on a positional controller which measures buffer levels. The controller is up- and down-clipped to meet strict QoS guarantees, while optimally distributing stochastically guaranteed and spared resources.

Simulation results based on ns-2 are presented to support the theory and the design, showing that the proposed scheduler is robust and performs always better than open loop scheduler in presence of traffic uncertainties.

I. Introduction

Wireless access through 802.11 W-LANs is spreading fast, becoming quickly as ubiquitous as cellular networks in places where access to the Internet is demanded.

While there are not hints that the use of the ISM 2.4 GHz spectrum is creating serious problems of interference and/or saturation, it is doubtlessly true that congestion within infrastructured Service Sets – i.e., "cells" served by an Access Point (AP), often spoils performances, specially of applications that require low latency. Even VoIP services work well on W-LANs, but their quality drops drastically as soon as traffic on-air becomes heavy or even just moderate [1]. Service support problems arise in all WLAN environments, but most of all in public HotSpots where customers pay for the service and specifically in networks conceived to support highly variable service demands, where dimensioning is more problematic, if at all possible.

IEEE 802.11 Task Groups have been at work to propose new solutions for the improvement of the Quality of Service (QoS) and service differentiation (IEEE 802.11 TGe), and more recently with 802.11 TGn to improve the throughput of the radio interface beyond the simple increase of the transmission speed, having recognized that the actual MAC protocol claims too high a toll on the scarce resources.

802.11 TGe completed its work on July 2005. The final document is now under revision for publication as IEEE standard, and we can expect the first standard-compliant devices to be on the market as early as mid 2006. 802.11 TGn work is still in an earlier stage, but it is already clear that the MAC and management part of the outcome of this TG will build upon 802.11e, making this latter an even more interesting solution.

This work was supported by the Italian Ministry for University and Research (MIUR) under the PRIN project TWELVE (http://twelve.unitn.it)

As we discuss shortly in Sect I-A, 802.11e defines a framework for the management of resources and traffic, but the actual algorithms and techniques used within the framework are open to competing implementations, and they can heavily affect the final performance obtained by a QAP (a QoS enabled Access Point) and the associated QSTAs (QoS enabled Stations). How much the research community is still interested — and working— on service provisioning in WLANs is testified by very recent surveys and position papers as [2], as well as scientific works discussed in Sect. I-B.

This paper addresses the problem of traffic scheduling by the QAP, discussing different possible implementations. The reference point, albeit naïve, is the simple scheduler (SS) drafted as example by the 802.11 TGe [3], which is conceived purely for CBR traffic, and obviously fails under any other condition. We propose and discuss two radically different possibilities: i) improving performances by a better characterization of the traffic, leading to open loop schedulers based on the notion of "*equivalent bandwidth*" (see Sect. II); and ii) improving performances by taking advantage of the feedback QSTAs send to the QAP describing the status of the local queue and applying closed-loop control techniques (see Sect. III).

In light of the results presented in Sect. IV, we argue that only the second choice represent a safe and robust implementation, and the overall complexity of the system, taking into account not only QAP computational requirement, but also QSTA "cooperation" and the interaction with existing applications, is indeed not larger that an open-loop solution with fixed allocation.

A. Overview and Modeling of 802.11e

The MAC protocol defined by 802.11e is the compromise between different needs: maintaining a backward compatibility, keeping the system complexity at bay, and enhancing its performance in terms of QoS support and differentiation. We refer to the Draft 11.0 [3], which is almost definitive and should not bear substantial differences with respect to the version submitted for Standard approval in July 2005.

The MAC protocol blends together an enhanced, QoS enabled version of the CSMA/CA used in the Distributed

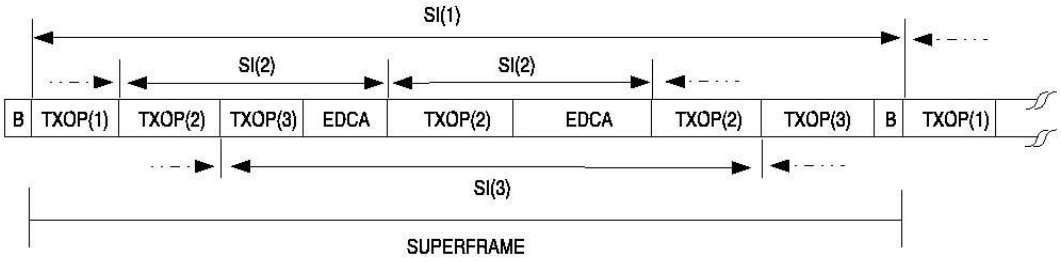


Fig. 1. General organization of the MAC protocol in 802.11e

Coordination Function (DCF) of 802.11, and named EDCA (Enhanced Distributed Channel Access), and a polling-based access named HCCA (HCF Controlled Channel Access[1]).

A plethora of solutions has been presented in recent years to differentiate services both in EDCA and HCCA. In this work we focus on scheduling within HCCA only, assuming that the EDCA period is used only for legacy stations, and is not used for QoS support or service differentiation. In Sect. I-B we discuss the literature that is most related with our work.

Fig. 1 describes the general evolution in time of the access procedure. The time is organized in superframes, which corresponds to the Beacon Interval (BI), i.e., the time between the transmission of two subsequent beacon frames. A superframe is divided into a contention free period (CFP), managed via HCCA, followed by a contention period (CP), where EDCA is used. The HCF, however, can interrupt the CP with periods, named Controlled Access Periods (CAP) when HCCA is used and the access is again contention free.

The HCF assigns resources to stations by allotting time in a Service Period (SP). An SP includes the time needed for the QAP to transmit frames to the QSTA and the Transmission Opportunity or TXOP for the QSTA. As shown in Fig. 2 TXOP assignment is done via polling. We identify the transmission opportunity of the i-th QSTA ad TXOP(i) and the service period for the same QSTA as SP(i). TXOP(i) represents the maximum time QSTA i can use the channel, including management and control frames. A QSTA that has nothing to transmit while its TXOP is still not expired should end it by transmitting a null frame.

QSTAs communicate their needs to the QAP on a per-connection basis (Traffic Stream – TS) and each QSTA can have up to eight parallel TSs.[2] The signaling and negotiation is done through the exchange of Traffic SPECifications (TSPEC), that contain (among others) the parameters relevant for traffic scheduling described in Table I. Besides the TSPEC, a QSTA sends the status of its internal queue to the QAP in the header of each data packet it transmits on-air. It is immediate to notice inspecting Fig. 1 and Table I that there are a number of open issues which must be addressed in implementing the HCF. For instance TXOPs are normally assigned on a per-station basis to spare resources wasted by multiple pollings to the same QSTA, while TSPEC are negotiated on a per-flow basis.

TSPEC parameter	Description
Nominal MSDU Size	Length (in octets) of MSDUs. One bit of this field indicates if MSDU size is constant or must be considered as "nominal"
Maximum MSDU Size	Maximum MSDU length in octets. The QAP should grant time to transmit at least one Maximum size MSDU per TXOP
Maximum Service Interval	Maximum admitted time between two consecutive polls (SI)
Minimum Data Rate	The scheduler is expected to allocate at least the time needed to serve this data rate
Mean Data Rate	The average traffic the TS generates
Peak Data Rate	The maximum data rate of the stream
Burst Size	Maximum amount of data that can arrive to the MAC-SAP with Peak Data Rate
Delay Bound	Maximum time a MSDU is allowed to queue before being successfully transmitted

TABLE I

TSPEC PARAMETERS RELEVANT FOR TRAFFIC SCHEDULING

QSTAs need not be polled in round robin, provided that consecutive pollings to the same station are not spaced more than an upper limit known as maximum Service Interval (SI) and negotiated between the QSTA and HCF. The service interval SI(i) of station i is the result of the different requirements of its traffic streams, but the HCF can also set it to a smaller value than needed, for instance to optimize the polling cycle. A very simple way to meet the requirements (though not necessarily the best in terms of performances) is setting all the SI identical one another and choose as common polling time the largest submultiple of the Beacon Interval which is smaller than the smallest required SI. This is the choice done in the simple scheduler SS.

Let k be a discrete time parameter indexing the successive polling cycles (PC). We define: $r_a(k)$ the vector representing the resources available during the k-th PC; $\overline{r}_i$ the mean (average) resources assigned to the i-th QSTA; $r_i(k)$ the resources actually assigned to the i-th QSTA during the k-th PC.

Notice that representing the system as a discrete time system there is no need to have all SI(i) equal one another: if station i is not polled at (discrete) time k, then $r_i(k) = 0$.

We assume that there is an admission control function that ensures stability of the system, so that for any meaningful

[1]HCF or Hybrid Coordination Function is the QAP entity that controls and manages the resources within the "cell."

[2]In the following we use the terms TS, connection, and flow interchangeably.

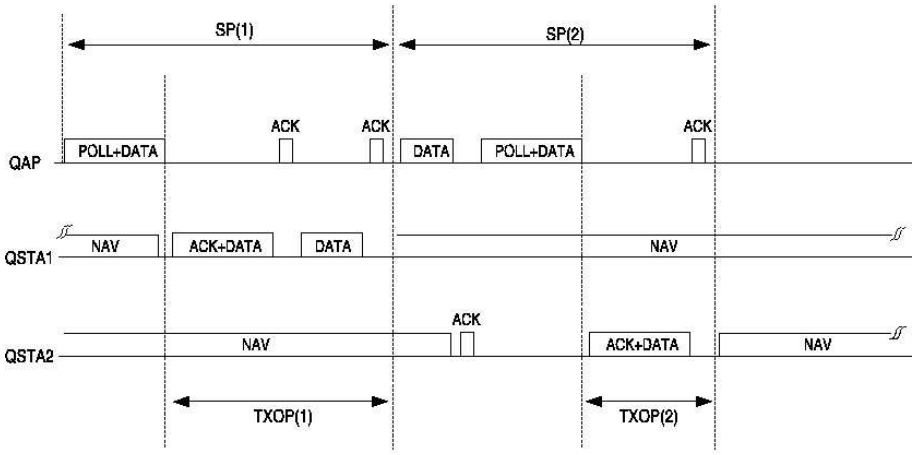


Fig. 2. Polling interaction between HCF and QSTA

observation interval K, given the number of associated QSTAs N_{QS}, the relationship

$$\frac{1}{K}\sum_{k=1}^{K} r_a(k) > \sum_{i=1}^{N_{\text{QS}}} \overline{r}_i \qquad (1)$$

holds and $\overline{r}_i$ can also be seen as the *guaranteed* resources assigned to station i. $r_a(k)$ is not necessarily constant over k, since PCs can have different length, due to non constant SIs, or to varying resources reserved for EDCA access (e.g., a non-QoS enabled STA associates or leave the QAP), etc.

From a scheduling point of view, we can assume that the 802.11e framework can be represented as in Fig. 3, where the the system a) represent the case of open-loop scheduling, and the system b) the case of closed-loop scheduling. The only difference between an open-loop and a closed-loop scheduler is whether the resource assignment function takes into account the remaining backlog of QSTAs $\text{bl}_i(k)$ at the end of the k-th SI or not. The block D_1 is a one-step delay representing the fact that the schedule defined by HCF during the k-th PC will be implemented by QSTAs during the next PC.

B. Related Work

There are many different proposals for managing the scheduling of resources in 802.11; we just mention here those that are closer to our approach, leaving aside all proposals centered on EDCA mechanisms.

The work in [4] is probably the closer to our work. The authors use a continuous time modeling, which is very accurate in analyzing performances, but is less prone of future optimization applying control-theoretical results.

The papers [5] and [6] discuss the use of variable service intervals trying to meet the deadlines of frames, the first one with an open-loop approach, and the second one accounting also for QSTAa feedbacks. Similar is also the approach described in [7], where an open-loop predictive scheduler is proposed. The prediction algorithm is based on measures of the actual traffic sent by TSs.

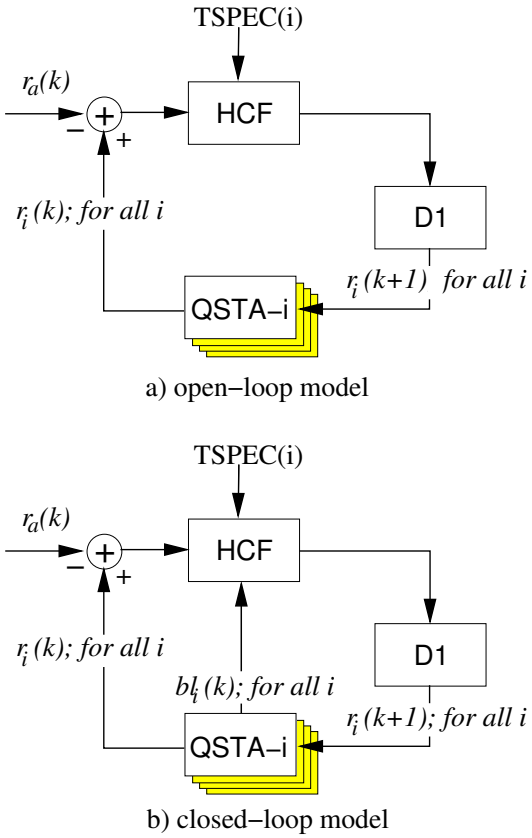


Fig. 3. Model of 802.11e scheduling: a) without using the feedback from QSTAs; b) using the feedback from QSTAs

Compared to all these works, the approach proposed in Sect. III is characterized by a faster dynamics of the closed loop scheduling adaptation, that leads to smaller delays in the channel access.

II. Equivalent Bandwidth Scheduling

The concept of "*Equivalent Bandwidth*" (EB) is not new. Indeed it dates back to the first CAC studies on VBR traffic.

The basic idea is quite simple. Given the stochastic characterization of an VBR source (e.g. a voice codec with Voice Activity Detection –VAD– and silence suppression) the EB of the source is the amount of resources r_{EB} that must be reserved to the source in order to guarantee that the amount of traffic T^s generated by the source will not exceed r_{EB} with a given probability p_o. Formally, given a source i

$$r_{\text{EB}}(i) : \mathbb{P}[T^s(i) > r_{\text{EB}}(i)] < p_o(i) \tag{2}$$

On the one hand, implementing an EB scheduler within the context of 802.11e scheduling is extremely easy. In fact, it is sufficient that the QSTA negotiate r_{EB} instead of the average bit rate in the reference SS of the standard draft and the QAP will regularly allocate the required resources. If the traffic generated by the flow is smaller than the negotiated r_{EB}, then the QSTA terminates TXOP(i) with the null frame and the spared resources are automatically freed for other stations' use. On the other hand, the main drawback of the equivalent bandwidth is not solved at all: if the QSTA traffic characterization is not precise, then r_{EB} will not meet the requirements of (2). Indeed, even if the source characterization is good, but it does not have Markovian properties, finding r_{EB} may not be an easy task. Finally, resource assignment in 802.11e is heavily quantized, since MPDU fragmentation is deprecated and inefficient. Quantization implies that the resources assigned will not match exactly the equivalent bandwidth of the flow, resulting in additional impairments.

We implemented (in ns-2) VBR sources as two- and three-states stochastic chains in order to to control the effect of approximated traffic characterization on the scheduler performance: if the transitions probabilities between chain states are geometric, then the chains are Markovian and (2) can be computed exactly. In most other cases (e.g., heavy tailed dwelling times, or even simple constant dwelling times) the equivalent bandwidth of the source is an approximation, and we expect the scheduler to have a worse performance in one way or the other, i.e., not meeting the requirements of the flow or performing poorly in accepting TSs. Fig 4 reports the discrete time chains implemented in ns-2. For each state s the user can choose the bit rate $R_b(s)$, the average frame dimension $D_f(s)$ and their distribution (e.g., constant, or truncated negative exponential), and the frame interarrival distribution.

The actual implementation is in form of a Discrete Time Chain with transitions upon packet transmission. The transition probabilities P_{ij} from state i to state j describe the behavior of the source every time a packet is generated. The transitions probabilities are computed so as to respect the average transmission rate of the source. For the three-state source some additional constraints may be required in order to have a unique solution.

III. Closed Loop Scheduling

As discussed in Sect. I-A QSTAs transmit their buffer level to the QAP. At the same time, the QAP is perfectly aware of

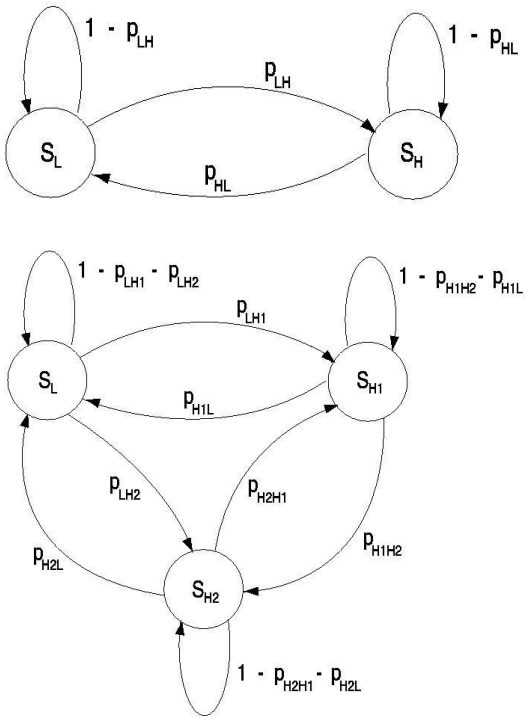


Fig. 4. Two- and three-state VBR models implemented in ns-2

the amount of traffic that has to be transmitted at the beginning of each SP to every QSTA.

Let's assume that the QAP knows the PHY transmission rates of QSTA, which is not unreasonable, since it can be assumed that transmission rates do not change from one SI to the next, so that the occasional transition from one rate to another can be accounted for as a disturbance in the control loop and it will be compensated automatically. In this situation the resource assignment can be done in bytes, and transformed in time to assign TXOPs taking into account the timing of the protocol, only at the end. Without loss of generality we can assume that the schedule is computed on a per-TS basis, and then the assignment is done on a per-QSTA basis in order to reduce the polling overhead.

The evolution of the transmission buffer B_j for each flow j is

$$B_j(k+1) = \max\left(0, [B_j(k) - r_j(k+1) + T_j^s(k+1)]\right)$$

If $B_j(k+1) > B_j^C$, where B_j^C is the buffer capacity for flow j, then $B_j(k+1) - B_j^C$ information is lost and $B_j(k+1) = B_j^C$. As stated with (1) we assume that the system is stable, so that the goal of any control technique that regulates r_j reduces to the minimization of some given metric $\mathcal{H}$ of the evolution in time of all the flows' buffers; first of all the loss probability.

There are many optimal control techniques that, given the metric $\mathcal{H}$ (e.g., $\mathcal{H}_2$ or $\mathcal{H}_\infty$) and the stochastic properties of the T^ss define the scheduling algorithm (see for instance [8]). At this stage of the research, however, we're more concerned with the fundamental properties of closed-loop scheduling,

rather than finding a theoretical optimal scheduler, which may turn out to be computationally complex, or loose its optimality properties due to implementation impairments. Therefore we resort to a basic positional controller (P-control) that tries to assign resources proportionally to the backlog. Since there is a guaranteed assignment to all flows, the assigned resources to flow j are

$$r_j(k) = r_j^{\min}(k) + r_j^{+}(k)$$

where $r_j^{+}(k)$ is a non-negative amount of additional resources assigned to flow j based on any weighted proportional function of the backlogs $B_j(k-1)$. Notice that $r_j^{\min}$ is a function of the discrete time k since SIs may be non constant.

Since we are dealing with a system that is intrinsically stable and inf-clipped (the buffer cannot be negative), we do not have to worry about stability of the controller, so that the repartition of resources among flows can also lead to a controller gain larger than one without affecting stability of the system.

A max-min fair proportional scheduler assigns additional resources based on the following proportionality

$$r_j^{+}(k+1) = \beta_j \frac{B_j(k)}{\sum_{j=1}^{N_{\text{TS}}} B_j(k)} \left[r_a(k) - \sum_{j=1}^{N_{\text{TS}}} r_j^{\min}(k) \right] \quad (3)$$

where β_j is a coefficient that can take into account flow priorities or any other differentiation policy within a given traffic class. The apportioning coefficient β_j can also be used to properly "weight," flows with different mean rates, since draining the same amount of information from different buffers takes a time which depends on the draining rate, so that the same backlog can result in different queuing delays. N_{TS} is the number of TS admitted by the CAC function.

Traffic classes can be easily taken into account either with a fixed priority scheme (i.e., the highest priority class is assigned additional resources first, then leftover resources are assigned to the second class, and so on) or with any other priority scheme that leads to the computation of r_j^{+} for the given class. Deriving the overall assignment scheme with a given priority enforcement is just cumbersome and is not reported here for the sake of clarity.

It must be noted that uplink and downlink flows are not identical, since the backlog information of uplink flows is one polling cycle old, because it reflects the situation at the QSTA when the last frame of the flow was transmitted on air, while the backlog of downlink flows can be known at the QAP without delays. We do not consider this problem anymore in this work, assuming that it does not introduce any meaningful bias.

The max-min based resource assignment (possibly distorted by the weights β_j that include the different mean rates), is based on the following considerations:

- The system is stable and backlogs are due to statistic fluctuations of the traffic sources (voice with VAD, video, etc.);
- The larger the backlog, the larger is the delay imposed to the the waiting information, and the larger is the probability that the flow buffer will overflow;
- If nothing is known about the stochastic processes driving the buffers, then the information loss probability is minimized when the buffers are all equalized.

A. *Closed-Loop scheduling with fixed* SI

The 802.11e draft [3] recommends using a fixed polling time for each station. Notice that, as depicted in Fig. 2, this does not necessarily mean that every SI is identical, but only that the same station is polled at fixed time intervals.

Implementing the proportional max-min fair P-controller defined by (3) with fixed SIs is straightforward and does not require any additional explanation[3].

We call this scheduler '*MaxMin Fair–Adaptive*' or MMF-A.

B. *Closed-Loop scheduling with dynamic oversampling*

The implementation described in Sect. III-A is perfectly complying with the draft. However, as already noted in [4], the reaction time of a closed loop scheduler with fixed SI can be as large as 2SI even for very low load conditions, which might penalize VBR sources, specially if sources have high variability. As we discuss presenting the results, the trivial solution of reducing SI is not practical, because reducing SI increases the overheads, thus penalizing the network under heavy load.

One possible solution is using dynamic SI values. This is not explicitly admitted by [3] (albeit neither explicitly forbidden), but we'll see that may significantly increase performances under rather normal operating conditions.

The discrete time theoretical framework depicted earlier in this paper remains identical also if SIs are changed dynamically. The problem is finding a way of changing SI dynamically. Fortunately, releasing the requirement of polling intervals to be constant, the solution is easy: if the k-th CFP ends at time $t(k)$ before a deadline $\tau(k)$ that defines a minimum guaranteed EDCA period before the next CFP, then the resources relative to the the time interval $\tau(k) - t(k)$ can be re-assigned with a new CAP, which defines a dynamic 'oversampling' of the controlled system. When the traffic fluctuations temporarily bring the system in overload, then $t(k) = \tau(k)$ and the normal SI intervals are used, so that global efficiency is preserved; when the system is not overloaded, but a few QSTA offer more traffic than their guaranteed share, the possibility of immediately re-assigning resources reduces the queue length, but most of all increases the probability that a frame will not violate the flow delay bound.

Since $\tau(k) - t(k)$ is normally rather small, assigning resources proportionally to the queue length is not possible due

[3]Indeed, the actual implementation requires a great deal of care to fulfill all standard draft requirements, and also due to the fact that resource assignments are quantized. These are however cumbersome details that do not add much to the fundamental idea explained so far. We refer the interested reader directly to the ns-2 implementation available on the TWELVE project website under the 'tools' menu (twelve.unitn.it/tools.html) — '802.11e closed-loop scheduling' — both for this scheduler and for all the others mentioned in this paper.

to quantization, so we decided to assign all of them to the source i for which

$$\max_{1<j<N_{\text{TS}}} \left(\beta_j \frac{B_j(k)}{\sum_{j=1}^{N_{\text{TS}}} B_j(k)} \right)$$

up to the equalization of its buffer with the one immediately smaller (in case of multiple stations with equal backlog values the one which is first in the polling schedule is selected).

We call this scheduler '*MaxMin Fair–Adaptive with Re-scheduling*' or MMF-AR.

Concluding this discussion of closed-loop scheduling, let's consider best effort traffic. It is common idea that the EDCA access scheme in 802.11 is the best one in supporting TCP-based best effort traffic, since polling schemes tend to be too rigid to adapt to the fast variability of best effort traffic. Indeed a closed-loop scheduler that reacts quickly to the presence of additional best effort traffic would spare the resources spent in collisions and backoffs of the EDCA protocol. There is however a major difference with respect to guaranteed traffic. In presence of greedy best effort sources, the stability of the system is not guaranteed with the traditional control-theory meaning, and the assignment of resources based on a P-controller of the source buffer obviously results in heavy unfairness. An example helps visualizing the situation. If two sources compete, but one starts before the other, say s_1 starts before s_2, then the transmission window of s_1 is much larger than the transmission window of s_2 when this latter starts transmitting. The buffer level at the QSTA will reflect the dimension of TCP transmission window. Any assignment scheme proportional to the buffer size, will favor s_1 and maintain the unfairness in time, unless there are losses and TCP reduces the congestion window size.

Indeed, to correct this bias, it is sufficient to apply a counting function

$$U_{B_i} = \begin{cases} 1 & \text{if } B_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

and evenly distribute the resources among stations that have some backlog.

We are currently evaluating the performance of the closed loop schedulers applied to best effort traffic, but the topic of best-effort traffic is not discussed further in this paper.

IV. Initial Results

We have implemented the schedulers and the VBR sources discussed in previous Sections in ns-2 [9] to evaluate the performance of our proposal. We consider five different possible schedulers: i) SS – the draft simple scheduler conceived for CBR traffic only; EB(0.2) and EB(0.01) – the same scheduler, but applied to TSPECs that reflect an ideal computation of the equivalent bandwidth of the VBR sources; MMF-A – the closed-loop scheduler with constant SI; MMF-AR – the closed loop scheduler with re-scheduling. The simple, open-loop scheduler allocates CFP and CAPs based on fixed SI intervals equal for all stations. The allocation is based only on the mean rate parameter of the TSPECs.

Mean bit rate = 128 kbit/s						
State	Param.	Value	Param.	Value	Param.	Value
L	R_b	64 kbit/s	D_f	120 bytes	T_{dw}	2.38 s
H	R_b	640 kbit/s	D_f	1200 bytes	T_{dw}	0.03 s

TABLE II

Parameters characterizing the two-state VBR source in the simulations; T_{dw} is the average dwell time in the state

Mean bit rate = 128 kbit/s						
State	Param.	Value	Param.	Value	Param.	Value
L	R_b	64 kbit/s	D_f	120 bytes	T_{dw}	0.36 s
$H1$	R_b	640 kbit/s	D_f	120 bytes	T_{dw}	0.009 s
$H2$	R_b	640 kbit/s	D_f	1200 bytes	T_{dw}	0.021 s

TABLE III

Parameters characterizing the three-state VBR source used in the simulations; T_{dw} is the average dwell time in the state

We have used both two- and three-state VBR sources, whose characterizing parameters are summarized in Tables II and III respectively. T_{dw} is the average time spent in the relative state.

For the sake of easy we only report results for homogeneous VBR sources competing for the uplink, leaving more complex scenarios including best effort traffic for further research. In this situation SI can be identical for all sources and we set it to 50 ms. Simulations have been run for 200 s of network operation or more.

Fig. 5 reports the loss probability performance of the five different scheduling schemes with non-delay sensitive sources. For these sources the delay bound of frames is set to ∞ and frame losses are only due to buffer overflows. The buffer size is measured in packets and is $B^C = 50$. Notice that this implies that the buffer size in bytes is variable depending on the size of frames it stores. Simulation points where no losses were recorded are not plotted.

The advantage of closed-loop scheduling with re-scheduling is evident in both plots, The other curves behavior is less straightforward to understand. With the simple two state sources, none of the other schedulers offer an acceptable behavior, even for highly underloaded networks where only a few flows are present. The reason lies in their impossibility to exploit unused resources, which are left for use to the EDCA access. Clearly letting all sources compete for resources during the EDCA phase would change the situation, but this is left for future research. MMF-AR, instead, re-scheduling unused resources at the end of polling cycle, is able to provide much better performance.

The results relative to the average transmission delay of packets, reported in Fig. 6, confirms the results, with the MMF-AR scheduler consistently obtaining lower transmission delays with respect to the others. We point out once more that fragmentation is inhibited in our results, and allowing fragmentation may change some results, allowing stations to exploit parts of the TXOPs where the whole packet to be

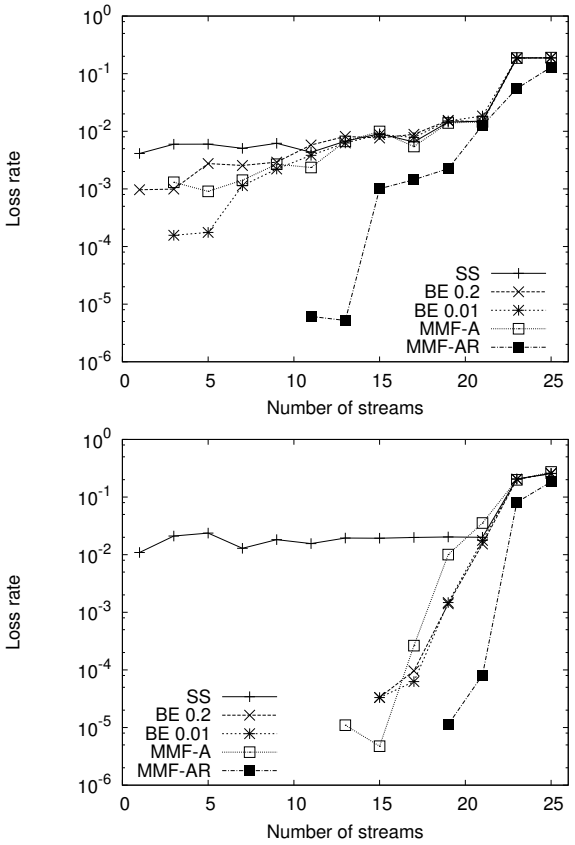


Fig. 5. Packet loss probability for the two-state (upper plot) and three-state (lower plot) VBR sources as a function of the number of concurrent flows for the five different considered schedulers for non-delay sensitive sources

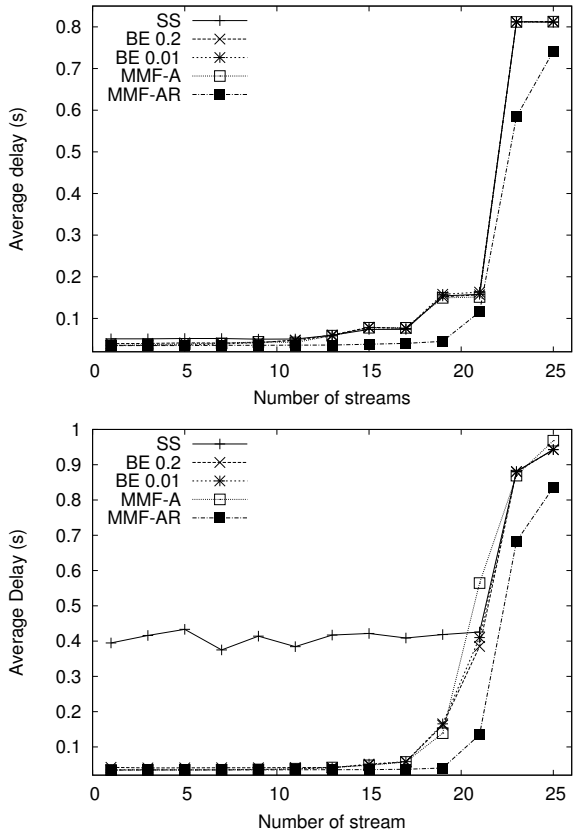


Fig. 6. Average frame delay for the two-state (upper plot) and three-state (lower plot) VBR sources as a function of the number of concurrent flows for the five different considered schedulers for non-delay sensitive sources

transmitted cannot be fit. However, we deem that finding a scheduler that performs well with variable size frames without requiring fragmentation is a major achievement, since fragmentation introduces overhead (in transmission and processing), and requires that both the sender and the receiver supports it.

We now restrict to study the more complex three-states sources for the sake of brevity. Fig. 7 refers to delay-sensitive sources, whose packets must be transmitted within 100 ms or they are discarded. This delay bound can be typical for voice of video-conferencing applications.

The behavior in this case is more easily interpreted. The SS and EB(0.2) schedulers make sources loose frames even in non-congested situations, simply because the resources allocated are fixed, and the VBR sources exceeds them. The loss rate in these cases can be theoretically computed starting from the VBR sources characteristics and the delay bound, and this computation confirms the simulation results. BE(0.01) and MMF-A behaves similarly, due to the delay in reaction of MMF-A, while the MMF-AR scheduler, reassigning unused resources to those flows that are currently above average obtains a performance that can be orders of magnitude better than the other schedulers. We point out here that the quantization of resources in the MMF-A and MMF-AR schedulers are slightly different and the one implemented

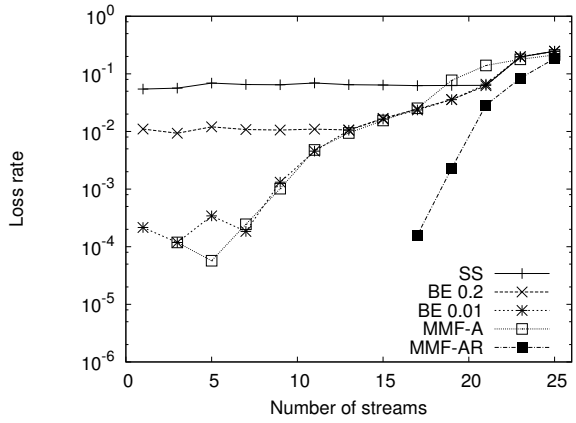


Fig. 7. Packet loss probability for the three-state VBR sources as a function of the number of concurrent flows for the five different considered schedulers for delay sensitive sources

in MMF-A is less efficient with large packets and high loads, which leads to the very bad behavior between 16 and 21 stations. We investigate this behavior in more detail at the end of the paper using real video sources.

As we already mentioned, the trivial solution to improve the performance of delay-sensitive traffic may seem the reduction of the SI. Fig. 8 reports the results with the same traffic configuration as Fig. 7, but with SI $=$ 25 ms. As correctly

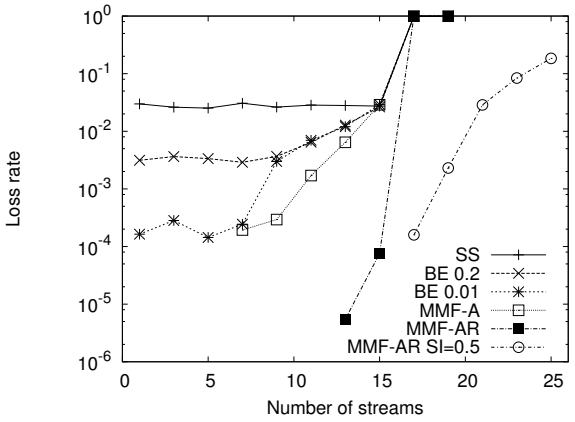


Fig. 8. Packet loss probability for the three-state VBR sources as a function of the number of concurrent flows for the five different considered schedulers for delay sensitive sources with SI= 25 ms

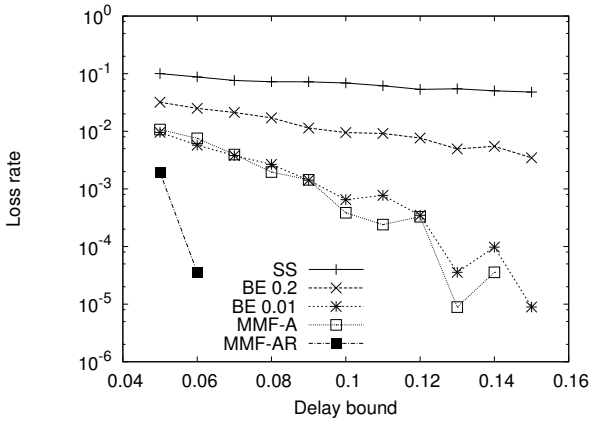


Fig. 10. Packet loss probability for the three-state VBR sources as a function the delay bound of frames with 8 concurrent flows for the five different considered schedulers

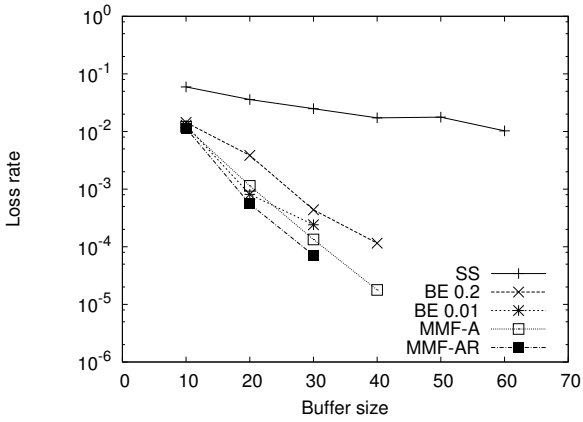


Fig. 9. Packet loss probability for the three-state VBR sources as a function the QSTA buffer size with 8 concurrent flows for the five different considered schedulers

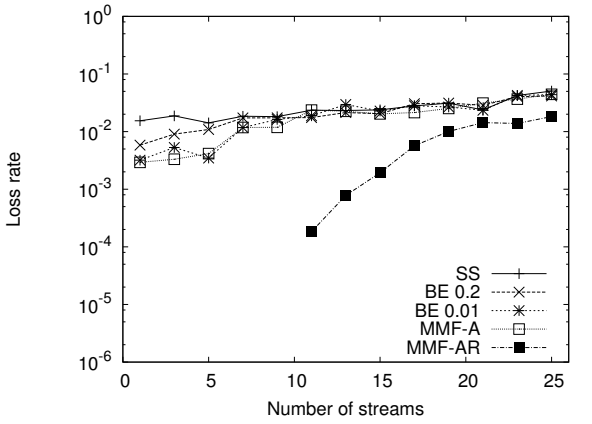


Fig. 11. Packet loss probability for the three-state VBR sources as a function the number of flows when the sources behaves differently from the declared TSPEC; non delay-sensitive sources

predicted in the analysis, the overheads of a shorter SI are dominant and it is very difficult to see any improvement, apart for very low loads. Indeed, comparing the two figures, it can be seen that all schedulers provide a smaller loss rate for very low loads, but as soon as the load increases the overheads, and the impossibility to assign the TXOPs to support a maximum size frame, makes the system performance unacceptable. The curve relative to MMF-AR with SI $= 50$ ms is reported in Fig. 8 for reference, showing that adapting the polling interval to the network conditions puts together the benefits of high load efficiency and low load performance.

To gain a better insight on the behavior, we analyze the sensitivity of the schedulers as a function of the buffer dimension (Fig. 9) and of the delay bound (Fig. 10). The behavior is consistent with theory, with the loss rate decreasing exponentially, but with different slopes depending on the scheduler efficiency. In both cases the MMF-AR scheduler performs consistently better than the other considered, while the MMF-A is always comparable to the EB(0.01), without requiring the complex (and unreliable) source characterization required to define the equivalent bandwidth.

We further analyze the performance in two non-standard, but realistic cases. Fig. 11 analyzes the behavior of the schedulers when the sources do not behave as declared in the TSPECs. Namely, the mean data rate is left unchanged, but the average time spent in the states is larger, so that high traffic bursts result longer. This result should be compared with Fig. 8. The performance loss of open-loop schedulers is evident, but also the MMF-A scheduler suffers, while the MMF-AR performance is far less influenced and remains acceptable. Fig. 12 explores what happens when sources are time sensitive, but the SIs supported by the QAP cannot be reduced. We set the delay bound to 1.5 the SI. All schedulers, apart from MMF-AR, have similar and unacceptable performances, with high losses even with only a few active flows. In particular MMF-A suffers from the fact that the delay bound is smaller than $2\times$SI, which is its response time.

Finally, Fig. 13 refers to simulations obtained with real video traffic[4] traces [10]. Since the stochastic descriptions

[4]The video traffic traces and the scripts to import them in ns-2 can be found at:
`http://www-tkn.ee.tu-berlin.de/research/trace/pub.html`

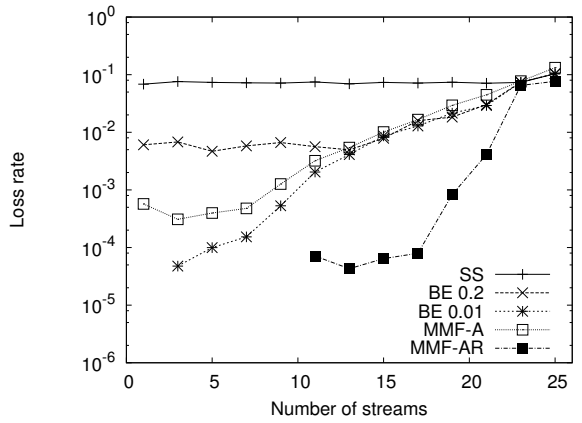


Fig. 12. Packet loss probability for the three-state VBR sources as a function of the number of flows with SI $= 100$ ms and delay bound 150 ms

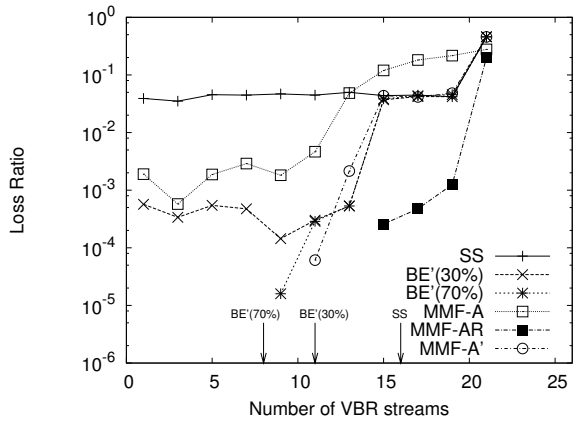


Fig. 13. Packet loss probability for real video traffic sources as a function the number of flows with SI $= 100$ ms and $DB = 150$ ms

of the sources is not available, we calculate EB' (we call it this way in order to distinguish it from EB previously defined in a more rigorous way) by considering as parameter the percentage of the difference between the peak and mean rates, and by adding such calculated value to the average rate. So, EB(0%) and EB(100%) corresponds to the mean and peak data rates respectively. The results show that SS is totally inefficient and MMF-A suffers quantization problems in adapting resources, due to the large average size of the MSDUs, while MMF-AR experiences better performances than all the others schedulers.

The curve labeled MMF-A' refers to the MMF-A scheduler with the quantization adopted by the MMF-AR scheduler. This enables to appreciate the behavior due to quantization phenomena from the one due to the SI flexibility.

The arrows labeled BE'(70%), BE'(30%), and SS reported on the x axis, indicate the CAC thresholds (resources nominally saturated) for the three open-loop schedulers. It is clear that BE'(30%) can guarantee the QoS, but only at the expenses of very low resources utilization. Finding a CAC surface for the closed-loop schedulers might be a difficult task; however, it is clear that simply using the declared mean data rate as in the SS scheduler guarantee a very good compromise between performance and resource utilization for the MMF-AR scheduler.

V. Discussion and Conclusions

This paper discussed different scheduler implementations to support QoS in 802.11e networks. In particular we argued about the use of open-loop, i.e., static and using only TSPEC information, or closed-loop, i.e., dynamic, using also information send back form the QSTAs schedulers.

Additionally we considered the possibility of using non-constant service intervals, showing that a close-loop scheduler with dynamic polling intervals trying to assign spare resources based on a P-controller on the buffer size can outperform other schedulers, including schedulers based on an Equivalent Bandwidth approach, and a closed-loop P-controller scheduler without dynamic polling.

The results we presented are not definitive, and additional research, for instance applying optimal control techniques or non-linear control techniques, is needed before the ideal 802.11e scheduler is found.

References

[1] Yang Xiao, Haizhon Li, "Voice and Video Transmissions with Global Data Parameter Control for the IEEE 802.11e Enhance Distributed Channel Access," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 1041–1053, Nov. 2004.

[2] N. Ramos, D. Panigrahi, S. Dey, "Quality of Service Provisioning in 802.11e Networks: Challenges, Approaches, and Future Directions," *IEEE Network*, pp. 14–20, July/August 2005

[3] "Medium Access Control (MAC) Enhancements for Quality of Service (QoS)," IEEE Draft Std 802.11e, D11.0, Oct. 2004.

[4] P. Ansel, Q. Ni, T. Turletti, "FHCF: An Efficient Scheduling Scheme for IEEE 802.11e," to appear in ACM/Kluwer MONET journal, Special issue devoted to WiOpt'04, 2005; Ext. version available as "FHCF: A Fair Scheduling Scheme for 802.11e WLAN," INRIA Research Report No. 4883, Sophia Antipolis, France, July 2003.

[5] A. Grilo, M. Macedo, M. Nunes, "A scheduling algorithm for QoS support in IEEE 802.11e networks," *IEEE Communication Magazine*, June 2003, pp. 36–43.

[6] D. Skyrianoglou, N. Passas, "Traffic Scheduling in IEEE 802.11e Networks Based on actual Requirements," In *Radio Network Management '04 (RNM'04)*, Athens, Greece, 2004.

[7] Lu Yang, "P-HCCA: A New Scheme for Real-time Traffic with QoS in IEEE 802.11e Based Networks," APAN Network Research Workshop 2004

[8] K. Zhou, J. C. Doyle, K. Glover, *Robust and Optimal Control*, Prentice Hall, Englewood Cliff, NJ, 1996.

[9] ns-2, network simulator (ver.2), Lawrence Berkeley Laboratory, URL: http://www-mash.cs.berkeley.edu/ns.

[10] Frank H.P. Fitzek, M. Reisslein, *MPEG-4 and H.263 Video Traces for Network Performance Evaluation (Extended Version)*, TU Berlin Dept. of Electrical Engineering, Telecommunication Networks Group, Technical Report: TKN-00-06, October 2000.

Queueing Delay Analysis of IEEE 802.11e EDCA

Paal E. Engelstad and Olav N. Østerbø

Abstract— **The majority of analytical work on the performance of IEEE 802.11 [1] focuses on predicting the throughput and the mean delay of only the medium access, although higher layer applications and protocols are interested in the total performance of the MAC layer. Seen in this perspective, surprisingly little focus has been on predicting the queueing delay. The main contribution of this paper opposed to other works is that it presents the full delay distribution through the z-transform. As a result, the mean medium access delay is found by the first order moment of the transform, and the mean queueing delay by the second order moment. Together this gives the average total delay associated with the MAC layer. The z-transform is derived from an analytical model that works in the whole range from a lightly loaded, non-saturated channel to a heavily congested, saturated medium. The model describes the priority schemes of the Enhanced Distributed Channel Access (EDCA) mechanism of the IEEE 802.11e standard [2]. EDCA provides class-based differentiated Quality of Service (QoS) to IEEE 802.11 WLANs, and distinguishes between four different traffic classes – called Access Categories (AC). By setting the number of ACs to one, and by using an appropriate parameter setting, the results presented are also applicable to the legacy 802.11 Distributed Coordination Function (DCF) [1]. The model predictions are calculated numerically and validated against simulation results. A good match between the analytical predictions and simulations was observed.**

Index Terms—**802.11e, Queueing Delay, Performance Analysis, EDCA, Z-transform of the Delay, Virtual Collision, Non-Saturation, AIFS, Starvation.**

I. INTRODUCTION

DURING recent years the IEEE 802.11 WLAN standard [1] has been widely deployed as the most preferred wireless access technology in office environments, in public hot-spots and in the homes. Due to the inherent capacity limitations of wireless technologies, the 802.11 WLAN easily becomes a bottleneck for communication. In these cases, the Quality of Service (QoS) features of the IEEE 802.11e standard [2] will be beneficial to prioritize for example voice and video traffic over more elastic data traffic.

The 802.11e amendment works as an extension to the 802.11 standard, and the Hybrid Coordination Function (HCF) is used for medium access control. HCF comprises the contention-based Enhanced Distributed Channel Access (EDCA) and the centrally controlled Hybrid Coordinated Channel Access (HCCA). EDCA has received most attention recently, and it seems that this is the WLAN QoS mechanism that will be promoted by the majority of vendors. EDCA is therefore the area of interest of this paper, and HCCA will not be discussed any further here.

EDCA allows for four different access categories (ACs) at each station and a transmission queue associated with each AC. Each AC at a station has a conceptual module responsible for channel access for each AC, and in this paper the module is referred to as a ”backoff instance”.

The majority of analytical work on the performance of 802.11e EDCA focuses on predicting the throughput, the frame dropping probabilities and the mean delay of the medium access. Surprisingly little focus has been on predicting also the queueing delay.

The importance of the queueing delay is evident. In realistic network scenarios, most of the MAC frames will carry a higher-layer packet, such as a TCP/IP or a RTP/UDP/IP packet, in the payload. A higher layer protocol or application will normally not interfere with the inner workings of the MAC layer. It might observe that it is subject to network delay (which is the case for TCP and many applications running on top of RTP), but it will normally not be able to distinguish between the types of delay. Thus, in most cases it is the total delay that counts. For analytical predictions of the delay of 802.11e EDCA to be useful, *both the queueing delay and the medium access delay should be considered*.

With little generated traffic (or with rate limiting e.g. in order to satisfy the Differentiated Services Expedited Forwarding Per-Hop-Behaviour) the mean queue length can be less than a packet, and the medium access delay is dominant. However, this case is not of the highest interest. First, the medium access delay then is typically less than a couple of milli-seconds (ms), and can normally be neglected compared to the comparably higher total end-to-end delay experienced for common Internet communication. Second, QoS analysis is not

Manuscript received September 19, 2005. This work was supported in part by the Open Broadband Access Network (OBAN) STREP project of the 6th Framework Program of the European Commission.

Paal E. Engelstad is with Telenor R&D, 1331 Fornebu, Norway (phone: +47 41633776; fax: +47 67891812; e-mail: paal.engelstad at telenor.com). He is also associated with “Universitetsstudiene på Kjeller” (UniK University Graduate Center).

Olav N. Østerbø is also with Telenor R&D, 1331 Fornebu, Norway (phone: +47 48212596; fax: +47 67891812; e-mail: olav-norvald.osterbo at telenor.com).

of very high interest with abundant channel resources. It is at the point when the channel becomes saturated that the differentiating features of 802.11e EDCA comes into play.

When the inter-arrival time of the generated traffic approaches the medium access delay, the queue begins to grow. It is observed that the medium access delay is still comparably low in this situation, while the queueing delay easily becomes dramatically higher. (The benefit of keeping the queue finite as a counter-measure is normally restricted, since a higher layer protocol is indifferent to whether the delay of the packet exceeds the limit for being useful or whether the packet is dropped in the queue.)

This paper presents a prediction of the mean queueing delay in addition to the mean medium access delay also predicted in earlier works. The z-transform of the delay is first found. This can provide all higher order moments of the delay. With the second order moment at hand, the mean queueing delay is easily derived. However, in order to derive the z-transform of the delay, an analytical model that also covers non-saturated channel conditions is first needed.

Most of the recent analytical work on the performance of 802.11e EDCA stems from the simple and fairly accurate model proposed by Bianchi [3] to calculate saturation throughput of 802.11 DCF. Later, Ziouva and Antonakopoulos [4] improved the model to find saturation delays, however, still of the undifferentiated DCF. They also improved the model by stopping the backoff counter during busy slots, which is more consistent with the IEEE 802.11 standard. Based on this work, Xiao [5] extended the model to the prioritized schemes provided by 802.11e by introducing multiple ACs with distinct parameter settings, such as the minimum and maximum contention window. Furthermore, this model also introduced finite retry limits. These additional differentiation parameters lead to more accurate results than previous models. (A list of references for other relevant efforts and model improvements of DCF can also be found in [5].)

We use a version of Xiao's model, however, extended as follow:

- The presented model predicts the performance not only in the saturated case, but in the whole range from a non-saturated medium to a fully saturated channel. (Some works, such as [6] and [7] have explored non-saturated conditions, however, only of the one-class 802.11. They are also primarily focussing on the non-saturation part instead of finding a good descriptive solution for the whole range.)
- In the non-saturation situation, our model accounts for "post-backoff" of an AC, although the queue is empty, according to the IEEE 802.11 standard. If the packet arrives in the queue after the "post-backoff" is completed, the listen-before-talk (or CSMA) feature of 802.11 is also incorporated in the model.
- Our model describes the use of AIFSN as a differentiating parameter, in addition to the other differentiation parameters encompassed by Xiao's efforts and other works.
- Virtual Collisions between the different transmission queues internally on a node are incorporated in the model.

The remaining part of the paper is organized as follows: The next section summarizes the differentiation parameters of 802.11e and provides the basis for understanding the analytical model. Section III presents the analytical model with AIFS differentiation and starvation prediction. Expressions for the throughput are first presented in Section IV (to give a complete presentation of the model), although it is the delay expressions presented in Section V that are the main contributions of this paper. The z-transform of the delay is first found. Then, the medium access delay is found as the first order moment of the transform. Finally, the queueing delay is found by means of the second order moment. In Section IV, the throughput expressions of the model are first validated against simulations to illustrate the accuracy of the model. Then, the mean access delay is validated, mainly because the prediction of the traffic intensity at which the queues grows to infinity, depends on it. By the end of the validation section, the queueing delay expression is validated. Our findings are finally summarized in the conclusions.

II. Differentiation Parameters of 802.11e

A. Selecting Contention Windows (CWs)

The traffic class differentiation of EDCA is based on assigning different access parameters to different ACs. First and foremost, a high-priority AC, i, is assigned a minimum contention window, $CW_{i,\min}+1$, and maximum contention window, $CW_{i,\max}+1$, that are lower than (or at worst equal to) that of a lower-priority AC.

For each AC, $i(i=0.,,,.3)$, let $W_{i,j}$ denote the contention window size in the j-th backoff stage i.e. after the j-th unsuccessful transmission; hence $W_{i,0}=CW_{i,\min}+1$. Let also $j=m_i$ denote the j-th backoff stage where the contention window has reached $CW_{i,\max}+1$. Finally, let L_i denote the retry limit of the retry counter. Then:

$$W_{i,j}=\begin{cases}2^j W_{i,0} & j=0,1,....,m_i-1\\ 2^{m_i} W_{i,0}=CW_{i,\max}+1 & j=m_i,....,L_i\end{cases} \quad (1)$$

B. Arbitration Inter-Frame Spaces (AIFSs)

Another important parameter setting is the Arbitration Inter-Frame Space (AIFS) value. When a backoff instance senses that the channel is idle after a packet transmission, it normally waits a guard time, AIFS, during which it is not allowed to transmit packets or do backoff countdown. Each AC[i] of 802.11e uses an Arbitration Inter-Frame Space (AIFS[i]) that consists of a SIFS and an AIFSN[i] number of additional time slots. In this paper A_i is defined as:

$$A_i = AIFSN[i] - AIFSN[N-1] \quad , \tag{2}$$

where N is the number of different ACs (i.e. normally four), and $AIFSN[N-1]$ is the AIFSN value of the highest priority AC, i.e. the lowest possible value. The 802.11e standard mandates that AIFSN[i] ≥ 2, where the minimum limit of 2 slots corresponds to the Distributed Interframe Space (DIFS) interval of legacy 802.11.

C. Transmission Opportunities (TXOPs)

Due to space limitations, priority based on differentiated Transmission Opportunity (TXOP) limits is not treated explicitly in this paper. Calculating the model with respect to different packet lengths and adjusting it to also cover contention-free bursting (CFB) is not difficult.

III. Analytical Model

A. The Markov Model

Figure 1 illustrates the Markov chain for the transmission process of a backoff instance of priority class i.

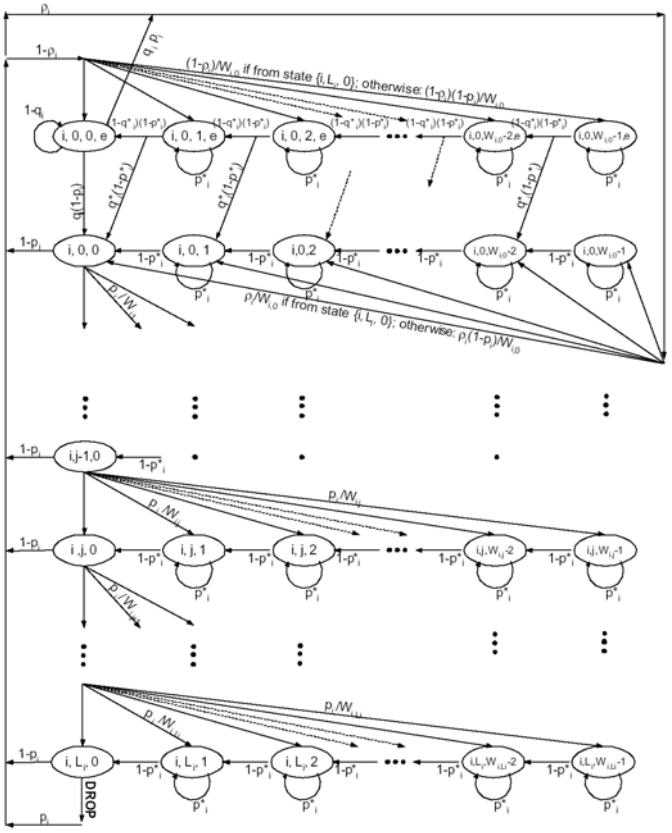


Figure 1. Markov Chain (both saturation and non-saturation)

In the Markov chain, the utilization factor, ρ_i, represents the probability that there is a packet waiting in the transmission queue of the backoff instance of AC i at the time a transmission is completed (or a packet dropped). Now, the backoff selects a backoff interval k at random and goes into post-backoff. If the queue is empty, at a probability $1-\rho_i$, the post-backoff is started by entering the state $(i,0,k,e)$. If the queue on the other hand is non-empty, the post-backoff is started by entering the state $(i,0,k)$. Hence, ρ_i balances the fully non-saturated situation with the fully saturated situation, and therefore plays a role to model the behaviour of the intermediate semi-saturated situation. When $\rho_i \to 1$ the Markov chain behaviour approaches that of the saturation case similar to the one presented by Xiao [5].

On the other hand, when $\rho_i \to 0$ the Markov chain models a stochastic process with a channel that is non-saturated. Then the backoff instance will always go into "post-backoff" after a transmission without a new packet ready to be sent.

While in the "post-backoff" states $(i,0,k,e)$ where $k > 0$, the probability that a backoff instance of AC i is sensing the channel busy and is thus unable to count down the backoff slot from one timeslot to the other is denoted with the probability p_i^*. If it has received a packet while in the previous state at a probability q_i^*, it moves to a corresponding state in the second row with a packet waiting for transmission. Otherwise, it remains in the first row with no packets waiting for transmission.

While in the state $(i,0,0,e)$, the backoff instance has completed post-backoff and is only waiting for a packet to arrive in the queue. If it receives a packet during a timeslot at a probability q_i, it does a "listen-before-talk" channel sensing and moves to a new state in the second row, since a packet is now ready to be sent. If the backoff instance senses the channel busy, at a probability p_i, it performs a new backoff. Otherwise, it moves to state $(i,0,0)$ to do a transmission attempt. The transmission succeeds at a probability $1-p_i$. Otherwise, it doubles the contention window and goes into another backoff.

For each unsuccessful transmission attempt, the backoff instance moves to a state in a row below at a probability p_i. If the packet has not been successfully transmitted after $L_i + 1$ attempts, the packet is dropped.

Let $b_{i,0,k,e}$ and $b_{i,j,k}$ denote the state distributions of the Markov chain. Since, the probability that transmission attempts enter stage j (where $j = 0,1,....,L_i$) is p_i^j, chain regularities yield:

$$b_{i,j,0} = p_i^j b_{i,0,0} \;;\; j = 0,1,...,L_i \,. \tag{3}$$

Furthermore, a backoff instance transmits when it is in any of the states $(i,j,0)$ where $j = 0,1,...,L_i$. Hence, if τ_i denotes the transmission probability (i.e. the probability that a backoff instance in priority class i transmits during a generic slot time, independent on whether the transmission results in a collision or not), it gives:

$$\tau_i = \sum_{j=0}^{L_i} b_{i,j,0} = b_{i,0,0} \frac{1-p_i^{L_i+1}}{1-p_i} . \quad (4)$$

Ways to express $b_{i,j,0}$ and p_i in terms of τ_i are presented in the following. Hence, a complete description of the system can be found by solving the above set of equations (one equation per AC i).

From chain regularities, and by working recursively through the chain from right to left in the upper row, it is seen that:

$$b_{i,0,k,e} = \frac{(1-\rho_i) b_{i,0,0}}{W_{i,0}(1-p_i^*)} \frac{1-(1-q_i^*)^{W_{i,0}-k}}{q_i^*} ;\ k = 1,2,...,W_{i,0}-1 . \quad (5)$$

Furthermore:

$$b_{i,0,0,e} = \frac{(1-\rho_i) b_{i,0,0}}{W_{i,0} q_i} \frac{1-(1-q_i^*)^{W_{i,0}}}{q_i^*} \quad (6)$$

and also:

$$b_{i,0,k} = \frac{W_{i,k}-k}{W_{i,0}(1-p_i^*)} (b_{i,0,0} + q_i p_i b_{i,0,k,e}) - b_{i,0,k,e}$$

$$\text{for } k = 1,2,...,W_{i,0}-1 . \quad (7)$$

The same analysis for the rest of the chain, gives:

$$b_{i,j,k} = \frac{W_{i,j}-k}{W_{i,j}(1-p_i^*)} p_i^j b_{i,0,0} \ ;\ j = 1,...,L_i, k = 1,...,W_{i,0}-1 . \quad (8)$$

Finally, normalization yields:

$$\frac{1}{b_{i,0,0}} = \sum_{j=0}^{L_i} \left[1 + \frac{1}{1-p_i^*} \sum_{k=0}^{W_{i,j}-k} \frac{W_{i,j}-k}{W_{i,j}} \right] p_i^j + \frac{1-\rho_i}{q_i} \frac{1-(1-q_i^*)^{W_{i,0}}}{W_{i,0} q_i^*} \left(1 + \frac{(W_{i,0}-1) q_i p_i}{2(1-p_i)}\right) . \quad (9)$$

The first sum in Eq. (9) represents the saturation-part, while the second part is the dominant term under non-saturation. Hence, the expression provides a unified model encompassing all channel loads from a lightly loaded non-saturated channel, to a highly congested, saturated medium. This full-scale model will be validated in Section VI

By performing the summations in Eq. (9) above and by assuming $m_i \le L_i$, Eq. (4) might be expressed as:

$$\frac{1}{\tau_i} = \frac{(1-2p_i^*)}{2(1-p_i^*)} +$$

$$\frac{W_{i,0}\left((1-p_i)(1-(2p_i)^{m_i}) + (1-2p_i)(2p_i)^{m_i}(1-p_i^{L_i-m_i+1})\right)}{2(1-p_i^*)(1-2p_i)(1-p_i^{L_i+1})} +$$

$$\left(\frac{1-p_i}{1-p_i^{L_i+1}}\right) \frac{1-\rho_i}{q_i} \frac{1-(1-q_i^*)^{W_{i,0}}}{W_{i,0} q_i^*} \left(1 + \frac{(W_{i,0}-1) q_i p_i}{2(1-p_i)}\right) \quad (10)$$

Eq. (10) is the key result in the analysis of the model. It represents a set of *N* equations that must be solved. They are normally inter-dependent in such a way that they must be solved numerically. However, there are cases, such as the one presented in [8], where a closed form solution can be found.

A. Estimating p_i without Virtual Collision Handling

The probability of unsuccessful transmission, p_i , from one specific backoff instance is given when at least one of the other backoff instances does transmit in the same slot. Thus,

$$p_i = 1 - \prod_{c=0, c\neq i}^{N-1} (1-\tau_c)^{n_c} = 1 - \frac{1-p_b}{1-\tau_i} \ , \quad \text{[without VC]}, \quad (11)$$

where p_b denotes the probability that the channel is busy (i.e. at least one backoff instance transmits during a slot time):

$$p_b = 1 - \prod_{i=0}^{N-1} (1-\tau_i)^{n_i} \ . \quad (12)$$

Furthermore, n_i denotes the number of backoff instances contending for channel access in each priority class i , and N denotes the total number of classes.

Eq. (11) is valid if each QSTA is transmitting traffic of only one AC and there are therefore totally $\sum_{i=0}^{N-1} n_i$ number of QSTAs transmitting traffic. Hence, no virtual collisions (VCs) will occur between different transmission queues on one QSTA.

B. Estimating p_i with Virtual Collision Handling

If each station is transmitting traffic of more than one AC, on the other hand, there *will* be virtual collision handling between the queues. Upon a virtual collision (VC) the higher priority AC will be attempted for transmission while the colliding lower priority traffic goes into backoff.

To illustrate this, consider that each QSTA is transmitting traffic of all *N* possible ACs, AC[*N*-1],..,AC[0]. In this paper AC[*N*-1] is by definition of the highest priority (normally equal to the "AC_VO" of 802.11e) and AC[0] of the lowest (normally equal to the "AC_BK" of 802.11e). The virtual collision handling implies that a backoff instance can transmit packets if other backoff instances don't transmit, *except* the backoff instances of the lower priority ACs *on the same QSTA*. Hence, instead of Eq. (11), , p_i is now found by:

$$p_i = 1 - \frac{1-p_b}{\prod_{c=0}^{i} (1-\tau_c)} \ , \quad \text{[with VC]} \quad (13)$$

where p_b is calculated as before [i.e. as in Eq. (12)].

B. Estimating p_i^* with Starvation Prediction

The reason for the distinction between p_i and p_i^* in the model is that AIFS-differentiation can be modelled with pretty good accuracy by adjusting the countdown blocking probability, p_i^* .

Lower priority backoff instances of class i have to suspend additional A_i slots after each backoff countdown. By assuming these are being smeared out randomly and distributed uniformly over all slots, it is possible to "scale down" the probability of detecting an empty slot, as illustrated in Figure 2.

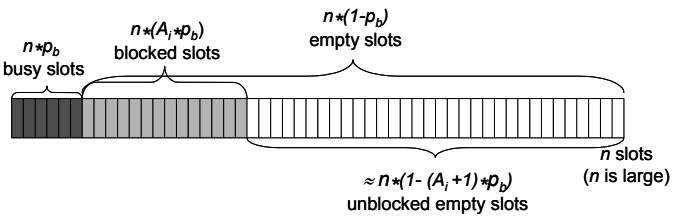


Figure 2. Simplified illustration of the principle of AIFS differentiation.

With this assumption, p_i^* can be approximated as:

$$p_i^* = \min(1, p_i + \frac{A_i p_b}{1-\tau_i}) \quad . \tag{14}$$

(The resemblance with Eq. (11) stems from the fact that the countdown blocking is not directly affected by the virtual collisions handling.)

Thus, starvation for AC i can be roughly predicted to occur when $p_i^* = 1$ or $p_b \geq \frac{1}{1+A_i}$ by Eq (14).

C. Estimating ρ_i

For a G/G/1 queue, the probability that the queue is non-empty, ρ is given by $\rho = \lambda \overline{D}$, where λ represents the traffic rate in terms of packets per second and $\overline{D}$ is the average service time. In this context $\overline{D}$ is the frame transmission delay from the time a packet has reached the front of the transmission queue and is the first packet to be transmitted until the packet is successfully transmitted or dropped.

For simplicity, here it is assumed that the traffic rate faced by all backoff instances of a class is the same on all stations, and use λ_i to denote the traffic rate (in terms of packets per seconds) of traffic class i on one station. Then;

$$\min(1, \lambda_i \overline{D}_i^{NON-SAT}) \leq \rho_i \leq \min(1, \lambda_i \overline{D}_i^{SAT}) \quad , \tag{15}$$

where $\overline{D}_i^{SAT}$ and $\overline{D}_i^{NON-SAT}$ are the delay with or without taking into account the post-backoff, respectively. It is correct to include the postback-off under full saturation and in this region $\overline{D}_i^{SAT}$ provides the best description of the delay. Under perfect non-saturation conditions, on the contrary, it is correct to omit the effects of post-backoff, and here $\overline{D}_i^{NON-SAT}$ provides the best delay description. The minimum bounds in Eq. (15) ensure that under saturation conditions, when the queue is always full of packets ready to be transmitted, the utilization, ρ_i, never exceeds 1. It is possible to use arguments to determine ρ_i with higher accuracy. Due to space limitations, this is beyond the scope of this paper.

Expressions for $\overline{D}_i^{SAT}$ and $\overline{D}_i^{NON-SAT}$ delays will be provided in Section IV.

D. Estimating q_i and q_i^*

To estimate q_i of the non-saturation model it is assumed that the traffic arriving in the transmission queue is Poisson distributed, i.e. that the system is of the M/G/1 type. q_i is the probability that at least one packet will arrive in the transmission queue during the following generic time slot under the condition that the queue is empty at the beginning of the slot.

Thus, q_i is calculated as:

$$q_i = 1 - \left(p_s e^{-\lambda_i T_s)} + (1-p_b) e^{-\lambda_i T_e} + (p_b - p_s) e^{-\lambda_i T_c} \right) . \tag{16}$$

We tested a number of different expressions for q_i^*, and observed that setting q_i^* equal to q_i for simplicity worked as a good approximation in all the scenarios explored.

IV. Throughput

Although the main focus of this paper is on the delay, the throughput predictions of the model is first presented. The reason is that it introduces some important probability definitions used also for the delay predictions later. Moreover, later in this paper the model is validated in terms of not only the delays, but also the throughput to give a more complete description of the accuracy of the model.

Let $p_{i,s}$ denote the probability that a packet from any of the n_i backoff instances of class i is transmitted successfully (at probability $\tau_i(1-p_i)$) in a time slot:

$$p_{i,s} = n_i \tau_i (1-p_i) . \tag{17}$$

where p_i is determined from Eq. (13) if there are virtual collisions [or Eq. (11) otherwise].

Let also p_s denote the probability that a packet from any class i is transmitted successfully in a time slot:

$$p_s = \sum_{i=0}^{N-1} p_{i,s} . \tag{18}$$

Then, the throughput of class i, S_i can be written as the average real-time duration of successfully transmitted packets by the average real-time duration of a contention slot that follows the special time scale of our model:

$$S_i = \frac{p_{i,s} T_{i,MSDU} B}{(1-p_b) T_e + p_s T_s + (p_b - p_s) T_c} \quad , \tag{19}$$

where T_e, T_s and T_c denote the real-time duration of an empty slot, of a slot containing a successfully transmitted packet and of a slot containing two or more colliding packets, respectively. The length of the longest colliding packet on the channel determines T_c. If all packets are of the same length, which will be considered in this paper, then $T_c = T_s$. (Otherwise refer to [3] to calculate T_c based on the average duration of the longest colliding data packet on the channel.) Finally, *B* denotes the nominal data bit-rate (e.g. 11 Mbps for 802.11b [9]), and $T_{i,MSDU}$ denotes the average real-time required transmitting the MSDU part of a data packet at this rate.

V. DELAY

A. Z-transform of the Medium Access Delay

We first deal with the delay associated with counting down backoff slots for the packets to be transmitted. While being blocked during countdown, the weighted average delay is $\frac{p_s}{p_b}T_s + (1-\frac{p_s}{p_b})T_c$, and the corresponding z-transform is:

$$D(z) = \frac{p_s}{p_b} z^{T_s} + (1-\frac{p_s}{p_b}) z^{T_c} . \quad (20)$$

While the backoff instance is counting down, the probability of facing an empty slot is $1-p_i^*$ while the probability of being blocked is p_i^*. Hence, the z-transform of this blocking delay is:

$$D_{bl}^i(z) = \frac{1-p_i^*}{1-p_i^* D(z)} . \quad (21)$$

When it is not blocked anymore, the system will spend an empty time-slot, T_e, when moving to the next countdown state. Hence, the z-transform of the total delay associated with one countdown state is:

$$D_{state}^i(z) = z^{T_e} D_{bl}^i(z) . \quad (22)$$

The total delay in a backoff stage is derived by a geometric sum over the probabilities associated with each countdown state:

$$D_{stage,j}^i(z) = \frac{1}{W_{ij}} \frac{1-(D_{state}^i(z))^{W_{ij}}}{1-D_{state}^i(z)} , \quad (23)$$

where the factor $1/W_{ij}$ reflects the uniform distribution of the selection of the number of backoff slots at each stage.

For simplicity the term $D_{level,j,s}^i(z)$ is introduced as:

$$D_{level,j,s}^i(z) = \prod_{l=s}^{j} D_{stage,l}^i(z) \quad (24)$$

Here, *s* is set to 0 under saturation conditions, because the post-backoff is undertaken before the transmission of each packet. Then the transform for the saturation delay may be written as:

$$D_{Sat}^i(z) = (1-p_i) \sum_{j=0}^{L_i} p_i^j z^{T_s + jT_c^*} D_{level,j,0}^i(z) + p_i^{L_i+1} z^{(L_i+1)T_c^*} D_{level,L_i,0}^i(z) . \quad (25)$$

Under extreme non-saturation conditions, on the contrary, the post-backoff is always completed before a new packet arrives in the transmission queue. Thus, under these conditions the post-backoff will not add to the transmission delay, as it did when the saturation delays were calculated above, and *s* is now set to 1 in Eq. (24). Then, the transform of the non-saturation delay can be found as:

$$D_{Non-Sat}^i(z) = (1-p_i) \sum_{j=0}^{L_i} p_i^j z^{T_s + jT_c^*} D_{level,j,1}^i(z) + p_i^{L_i+1} z^{(L_i+1)T_c^*} D_{level,L_i,1}^i(z) , \quad (26)$$

where $D_{level,0_i,1}^i(z) = 1$ has been defined for convenience.

The first part of Eq. (25) and of Eq. (26) represent the delay associated with packets that are eventually transmitted successfully on the channel, where p_i is the probability of colliding after each *j*-th stage, adding an extra delay of T_c^* (thus the factor $z^{T_c^*}$ per stage). $(1-p_i)$ is the probability of finally transmitting the packet after a stage, which adds an extra delay of T_s (thus the factor z^{T_s}). The last part of Eq. (25) and of Eq. (26) represent the delay of packets that go through all $0,...,L_i$ stages without being transmitted successfully, and are eventually dropped.

B. Mean Medium Access Delay

Finally, the mean medium access delay when the post-backoff delay is taken into account, $\overline{D}_i^{SAT}$, is found directly from the transform in Eq. (25):

$$\overline{D}_i^{SAT} = D_{Sat}^{i\,(1)}(1) = (1-p_i^{L_i+1})(T_s + T_c^* \frac{p_i}{1-p_i}) + \frac{\overline{D}_i^{state}}{2} R_1^i \quad (27)$$

where $\overline{D}_i^{state}$ is defined as the mean delay associated by a countdown state:

$$\overline{D}_i^{state} = D_{state}^{i\,(1)}(1) = T_e + \left[\frac{p_s}{p_b} T_s + (1-\frac{p_s}{p_b}) T_c \right] \frac{p_i^*}{(1-p_i^*)} \quad (28)$$

and the sum R_1^i is given by:

$$R_1^i = \sum_{j=0}^{L_i} p_i^j (W_{ij} - 1) . \quad (29)$$

By performing the summation above in Eq (25) for the case $m_i \le L_i$ the following explicit expression for R_1^i is obtained:

$$R_1^i = W_{i0}\left(\frac{1-(2p_i)^{m_i+1}}{1-2p_i} + 2^{m_i}\,\frac{p_i^{m_i+1} - p_i^{L_i+1}}{1-p_i}\right) - \frac{1-p_i^{L_i+1}}{1-p_i} \quad . \tag{30}$$

The mean medium access delay when the post-backoff delay is not taken into account, $\overline{D}_i^{NON-SAT}$, can be calculated similarly using Eq. (26), or it may alternatively be found by:

$$\begin{aligned}\overline{D}_i^{NON-SAT} &= \left[\frac{D_{Sat}^i}{D_{Stage,0}^i}\right]^{(1)}(z=1)\\ &= \frac{D_{Sat}^{i\,(1)}(1)D_{Stage,0}^i(1) - D_{Sat}^i(1)D_{Stage,0}^{i\,(1)}(1)}{D_{Stage,0}^i(1)}\\ &= D_{Sat}^{i\,(1)}(1) - D_{Stage,0}^{i\,(1)}(1) = \overline{D}_i^{SAT} - \overline{D}_i^{state}\frac{W_{i0}-1}{2}\end{aligned} \tag{31}$$

where $\overline{D}_i^{state}$ is given in Eq.(28). The resolution of $D_{stage,j}^{i\,(1)}(1)$, shown by the last equality, is found by simple derivation of Eq. (23) and subsequent application of L'Hôpital's rule three times.

C. Mean Queueing Delay

By considering the medium access delay as the "service time" for a packet in an single server queue we may obtain the mean queueing delay by applying the corresponding formula for the M/G/1 queueing model, $\overline{\Delta_i}$; given through the second order moment of the delay [10]:

$$\overline{\Delta_i} = \frac{\lambda_i \overline{D_i^2}}{2(1-\rho_i)} \quad . \tag{32}$$

To apply an M/G/1 model for the queueing delay we must also assume that the medium access times are independent stochastic variables. This will not be an exact assumption, however, it is believed that the dependencies will be weak, so that Eq. (32) will provide an accurate approximation.

We will first consider the queueing delay when effects of the post-backoff delay are taken into account, which gives the best description close to saturation conditions. Later in this section we deal with the opposite case, which better describes the non-saturation situation. The second order moment of the delay is found by derivation of the z-transform [10]:

$$\begin{aligned}\overline{D_i^2}^{SAT} &= \overline{D}_i^{SAT} + D_{Sat}^{i\,(2)}(1) = (1-p_i)\sum_{j=0}^{L_i}(T_s + jT_c^*)^2 p_i^j +\\ &2(1-p_i)\sum_{j=0}^{L_i}(T_s + jT_c^*)p_i^j\,\overline{D_{level,j,0}^i} + (1-p_i)\sum_{j=0}^{L_i}p_i^j\,\overline{D_{level,j,0}^i}^2 +\\ &((L_i+1)T_c^*)^2 p_i^{L_i+1} + 2(L_i+1)T_c^* p_i^{L_i+1}\overline{D_{level,L_i,0}^i} + p_i^{L_i+1}\overline{D_{level,L_i,0}^i}^2\end{aligned} \tag{33}$$

where

$$\overline{D_{level,j,0}^i} = D_{level,j,0}^{i\,(1)}(1) = \overline{D}_i^{state}\sum_{l=0}^{j}\frac{W_{il}-1}{2} \tag{34}$$

and

$$\begin{aligned}\overline{D_{level,j,0}^i}^2 &= D_{level,j,0}^{i\,(2)}(1) + D_{level,j,0}^{i\,(1)}(1) =\\ &\left(\overline{D}_i^{state}\right)^2\left(\left(\sum_{l=0}^{j}\frac{W_{il}-1}{2}\right)^2 - \sum_{l=0}^{j}\left(\frac{W_{il}-1}{2}\right)^2 + \sum_{l=0}^{j}\frac{(W_{il}-1)(W_{il}-2)}{3}\right) +\\ &\overline{D_i^{state}}^2\sum_{l=0}^{j}\frac{W_{il}-1}{2}\end{aligned} \tag{35}$$

Furthermore, $\overline{D}_i^{state}$ is given by Eq.(28) and

$$\begin{aligned}\overline{D_i^{state}}^2 &= D_{state}^{i\,(2)}(1) + D_{state}^{i\,(1)}(1) =\\ &T_e^2 + \left[\frac{p_s}{p_b}T_s(2T_e+T_s) + (1-\frac{p_s}{p_b})T_c(2T_e+T_c)\right]\frac{p_i^*}{1-p_i^*} +\\ &2\left[\frac{p_s}{p_b}T_s + (1-\frac{p_s}{p_b})T_c\right]^2\left(\frac{p_i^*}{1-p_i^*}\right)^2\end{aligned} \tag{36}$$

By performing the summations in the expressions above we obtain:

$$\begin{aligned}\overline{D_i^2}^{SAT} &= \left(1-p_i^{L_i+1}\right)\left(T_s^2 + T_c^{*2}\frac{p_i}{1-p_i}\right) +\\ &2T_c^*\left(1-(L_i+1)p_i^{L_i} + L_i p_i^{L_i+1}\right)\left(T_s + T_c^*\frac{p_i}{1-p_i}\right)\frac{p_i}{1-p_i} +\\ &\overline{D}_i^{state}\left[\left(T_s + T_c^*\frac{p_i}{1-p_i}\right)\left(R_i^1 - p_i^{L_i+1}R_2^i\right) + T_c^* R_i^3\right] +\\ &\left(\overline{D}_i^{state}\right)^2\left(\frac{R_4^i}{3} + \frac{R_5^i - R_3^i}{2}\right) + \overline{D_i^{state}}^2\frac{R_1^i}{2}\end{aligned} \tag{37}$$

where the sum R_1^i is defined by Eq. (28) and the other sums $R_2^i, \ldots, R_5^i$ are defined by:

$$R_2^i = \sum_{j=0}^{L_i}(W_{ij}-1) \quad , \tag{38}$$

$$R_3^i = \sum_{j=1}^{L_i} j p_i^j (W_{ij}-1) \quad , \tag{39}$$

$$R_4^i = \sum_{j=0}^{L_i} p_i^j (W_{ij}-1)(W_{ij}-2) \quad , \tag{40}$$

$$R_5^i = \sum_{j=1}^{L_i} p_i^j (W_{ij}-1)\sum_{s=0}^{j-1} W_{is} \quad . \tag{41}$$

Explicit expressions for the sums $R_2^i, \ldots, R_5^i$ in Eq. (38) – Eq. (41) can be found by performing the summation for the case $m_i \le L_i$. These explicit expressions are found in the Appendix of this paper.

One may make the same type conversion between $\overline{D_i^2}^{SAT}$ and $\overline{D_i^2}^{NON-SAT}$ using a similar approach as in Eq. (31):

$$\overline{D_i^2}^{NON-SAT} = \overline{D_i^2}^{SAT} - \overline{D_{stage,0}^i}^2 - 2\overline{D}_i^{NON-SAT}\,\overline{D_{stage,0}^i} \quad . \tag{42}$$

where

$$\overline{D_{stage,0}^i} = D_{stage,0}^{i\,(1)}(1) = \overline{D_{state}^i}\frac{W_{i0}-1}{2} \quad , \tag{43}$$

and

$$\overline{D_{stage,0}^{i}{}^{2}} = D_{stage,0}^{i}{}^{(2)}(1) + D_{stage,0}^{i}{}^{(1)}(1) = \left(\overline{D_i^{state}}\right)^2 \frac{(W_{i,0}-1)(W_{i,0}-2)}{3} + \overline{D_i^{state^2}} \frac{(W_{i,0}-1)}{2}, \quad (44)$$

and where $\overline{D_i^{state}}$ and $\overline{D_i^{state^2}}$ are given by Eq.(28) and Eq.(36). Furthermore, $\overline{D_i}^{NON-SAT}$ and $\overline{D_i^2}^{SAT}$ are given by Eq.(31) and Eq.(37).

VI. VALIDATIONS

A. Simulation Setup

We compared numerical computations in *Mathematica* with ns-2 simulations, using the TKN implementation of 802.11e [11] for the ns-2 simulator.

The scenario selected for validations is 802.11b with long preamble and without the RTS/CTS-mechanism.. The parameter settings for 802.11b are found in [9]. Based on these, the model parameters $T_e = 20\mu s$, $T_{i,MSDU} = T_{1024} = 520\mu s$ and $T_S = T_c = 1321\mu s$ were estimated. Finally, setting the time a colliding station has to wait when experiencing collision, T_c^*, equal to the time a non-colliding station has to wait when observing a collision on the channel, T_c, corresponds with the ns-2 implementation used for validations.

Parameters such as CWmin and CWmax are overridden by the use of 802.11e [2]. For the validations, the default 802.11e values, also shown in Table 1 in [8], were used.

The node topology of the simulation uses five different stations, QSTAs, contending for channel access. Each QSTA uses all four ACs, and virtual collisions therefore occur. Poisson distributed traffic consisting of 1024-bytes packets was generated at equal amounts to each AC.

The throughput values of our ns-2 simulations were measured over 3 minutes of simulation time. The simulations were started with a 100 seconds transition period to let the system stabilize before the measurements were started.

B. Validation of the Throughput Predictions

Although the main focus of this paper is on the delay, the throughput predictions of the model is first validated, in order to give a more complete impression of the accuracy of the model that is being used.

Figure 3 compares numerical throughput calculations of the analytical model with the actual simulation results. It is observed that the model corresponds relatively well with the outcome of the simulations. However, there are some differences that exceed the 95% confidence interval of the simulations. (Since the intervals are so small they have only been shows for 3000 Kbps and 5000 Kbps in Figure 3).

We also see that the starvation of AC[0] and AC[1], experienced with simulations, is described with relatively good accuracy by the analytical model. However, the starvation expression in Eq. (14) seems to be a little to coarse-grained to model the exact throughput behavior when these ACs face starvation. In the semi-saturation-part (middle part) of the figure it is also observed some inaccuracies in the numerical calculations of model. *Mathematica* have difficulties in converging in this region, for example when the traffic generated per AC is around 2500 Kbps.

C. Validation of the Medium Access Delay Predictions

Even in all the cases where the queueing delay is significantly higher than the medium access delay, the latter is not unimportant. It is the medium access delay that determines whether the service rate of the MAC is able to match the traffic rate that enters the queue. For this reason, the medium access predictions are validated first.

Figure 4 compares numerical mean delay calculations of the analytical model with the actual simulation results. The solid marked with triangles show the numerical results for the mean saturation delay, $\overline{D}_i^{SAT}$, i.e. the delay that includes the post-backoff. The dotted curves marked with triangles show the mean non-saturation delay, $\overline{D}_i^{NON-SAT}$, i.e. the delay that does not take into account the effects of the post-backoff. Ideally, the mean delay, $\overline{D}_i$, of each AC *i* (represented by dashed curves marked with 'X's in Figure 4) should lie between the two numerically calculated curves for $\overline{D}_i^{NON-SAT}$ and $\overline{D}_i^{SAT}$:

$$\overline{D}_i^{NON-SAT} \le \overline{D}_i \le \overline{D}_i^{SAT}. \quad (45)$$

We observe that this is the case in most parts of the figure. However, the model predicts a delay for the second highest priority AC, AC[2], that is slightly lower than experienced by the simulations around 3000 Kbps. The 95% confidence interval for AC[2] - drawn at 3000 Kbps in Figure 4 - shows that this discrepancy cannot be explained by simple statistical variations. (The 95% confidence intervals are also shown for 5000 Kbps.)

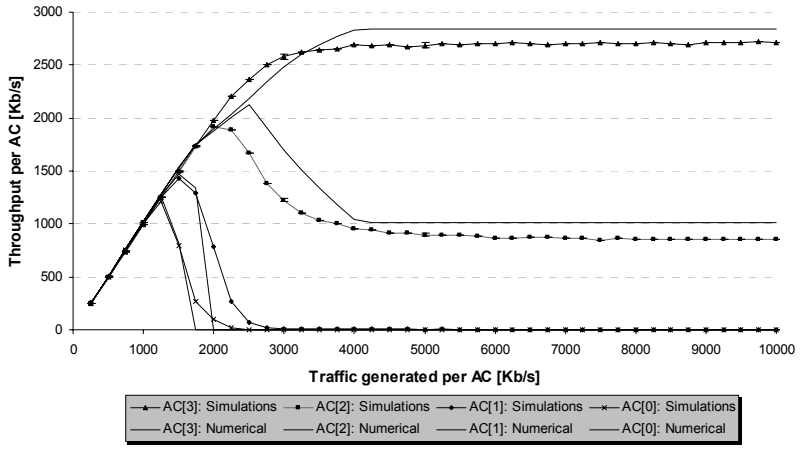


Figure 3. Throughput comparison between analytical (numerical) and simulation results with four ACs per station and varying traffic per AC.

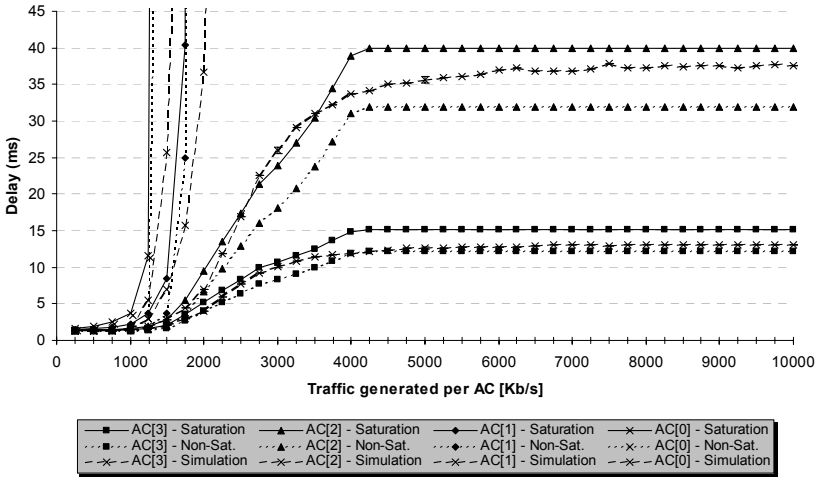


Figure 4. Mean Medium Access Delay comparison between analytical (numerical) and simulation results with four ACs per station and varying traffic per AC. (The "Saturation" curves refer to the delay calculated when the effects of the post-backoff delay are taken into account, while for the "Non-Sat." curves, these effects are not considered.)

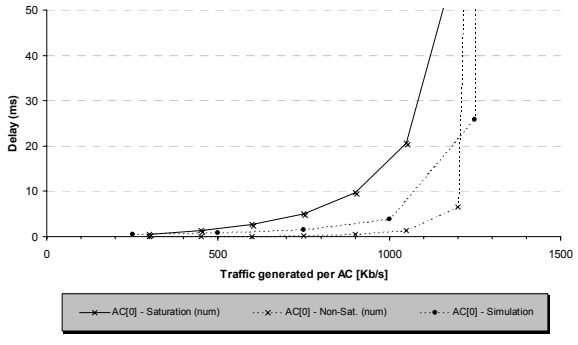


Figure 5. Mean Queueing Delay comparison of AC[0] between analytical (numerical) and simulation results.

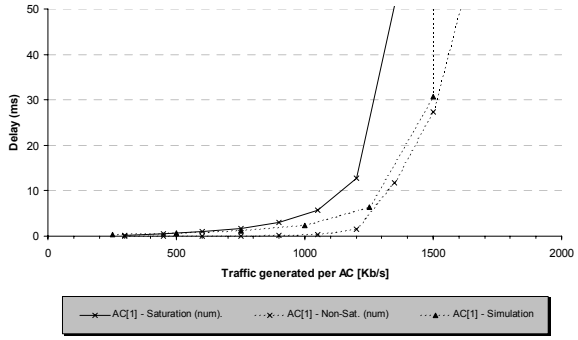


Figure 6. Mean Queueing Delay comparison of AC[1] between analytical (numerical) and simulation results.

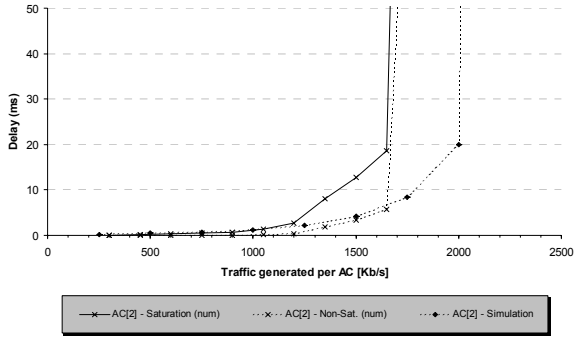


Figure 7. Mean Queueing Delay comparison of AC[2] between analytical (numerical) and simulation results.

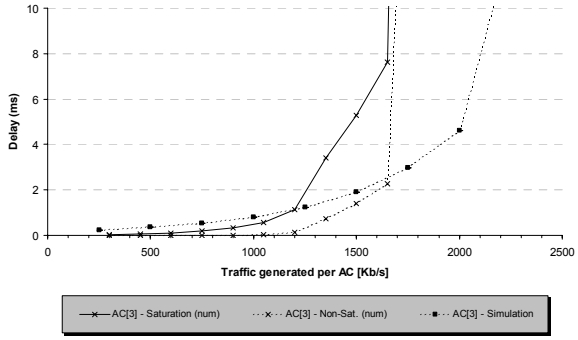


Figure 8. Mean Queueing Delay comparison of AC[3] between analytical (numerical) and simulation results.

Furthermore, the model predicts that the delay for the lowest priority ACs, AC[0] and AC[1], increases to infinity a little faster than observed in the simulations. This result corresponds well with the inaccuracies seen for the throughput in the corresponding region in Figure 3.

D. Validation of the Queueing Delay Predictions

The predicted mean queueing delay (numerically calculated) is compared with simulation results for the same 5-station scenario above. As argued for earlier, the mean queueing delay, $\overline{\Delta}_i$, should lie between the two numerical predictions, $\overline{\Delta}_i^{SAT}$ and $\overline{\Delta}_i^{NON-SAT}$, depending on whether the effects of the post-backoff delay are taken into account or not:

$$\overline{\Delta}_i^{NON-SAT} \le \overline{\Delta}_i \le \overline{\Delta}_i^{SAT} . \quad (46)$$

Figure 5, Figure 6, Figure 7 and Figure 8 show the queueing delay comparisons for AC[0], AC[1], AC[2] and AC[3], respectively. It is observed that the simulation results are largely within the prediction range given by Eq. (46).

However, close to the saturation singularity where queues grow infinitely, it seems that the model is less accurate, and the simulation results are outside this range (except for AC[0] in Figure 6). The most important reason is probably that inaccuracies in the mean delay directly affect the exact location on the absissa axis (x-axis) where this singularity occurs. This is seen directly from Eq. (32), since in the nominator ρ_i is determined by $\rho_i = Max[1, \lambda_i \overline{D}_i]$. Hence, the prediction of whether the system has reached the saturation requirement, $\rho_i = 1$, or not at a given traffic intensity, λ_i, is fully dependent on the size of $\overline{D}_i$. Small inaccuracies in the prediction of $\overline{D}_i$ can translate into large inaccuracies in the prediction of the exact traffic intensity where the singularity will occur.

VII. CONCLUSIONS

This paper demonstrates the importance of the queueing delay, and shows how analytical models can predict it. Using a saturation model makes no sense, since the queueing delay is infinite under saturation conditions. Instead, a model that has been extended to cover the full range from a non-saturated to a fully saturated channel is used. Furthermore, a simple way to introduce AIFS differentiation into the model is proposed. Thus, the default medium access parameters recommended by the 802.11e specification, which uses this kind of differentiation, can be studied.

Earlier works (such as [5]) have mostly focused on mean values for the medium access delay. Here, however, all higher-order moments of the delay are found, through the explicit z-transform of the delay. The average queueing delay can then be predicted by means of the second order moment of the delay transform, as a direct consequence of basic queueing theory. The mean medium access delay and the mean queueing delay together, constitute the average total delay of the MAC, as seen from an upper layer protocol or application.

The mean queueing delay predictions of the model are calculated numerically and validated against simulations. The mean access delay was also validated, since it has a direct impact on when saturation occurs and when queues as a result grow to infinity. To make the analysis complete, validations of the throughput were also presented.

It is observed that the predictions of the mean queueing delay give a relatively good match with simulations. The mean access delay and the throughput were also relatively well predicted by the model.

It is finally pointed out that expressions for the mean queueing delay and mean access delay were found under two extremities of the model, namely whether or not the effects of the post-backoff queuing delay were taken into account (depending on whether one wants to find the delay close to saturation or under non-saturation conditions). In fact, the z-transform was given in terms of these two limits. Thus, the delay predictions presented here say that the real mean delay values must lie somewhere between these limits. The presented model, however, contains parameters (such as q_i^* shown in Figure 1) that should make it feasible to derive a more exact expression. As a first order approximation, the following could be attempted:

$$\begin{aligned} \overline{D}_i &\approx (1-\rho_i)\overline{D}_i^{NON-SAT} + \rho_i \overline{D}_i^{SAT}; \\ \overline{\Delta}_i &\approx (1-\rho_i)\overline{\Delta}_i^{NON-SAT} + \rho_i \overline{\Delta}_i^{SAT} . \end{aligned} \quad (47)$$

These and more exact expressions will be explored in a follow-up paper.

This paper presents the medium access delay distribution through the z-transform. In addition to finding the moments of the delay, the z-transform can be inverted numerically with a configurable error bound. By assuming an M/G/1 queueing model it is possible to obtain a complete delay description, containing the distributions both of the MAC delay, the queueing delay and the total delay. All desirable delay percentiles follow. This follow-up work will be published and presented soon [12].

APPENDIX

The explicit expressions for the sums $R_2^i,\ldots,R_5^i$ of Eq. (38) – Eq. (41) can be found by performing the summation for the case $m_i \le L_i$:

$$R_2^i = W_{i0}\left(2^{m_i}(2+L_i-m_i)-1\right)-(L_i+1) \quad , \tag{48}$$

$$R_3^i = W_{i0}\left(\frac{2p_i\left(1-(m_i+1)(2p_i)^{m_i}+m_i(2p_i)^{m_i+1}\right)}{(1-2p_i)^2}+ 2^{m_i}\,\frac{p_i\left((m_i+1)p_i^{\,m_i}-m_i p_i^{\,m_i+1}-(L_i+1)p_i^{\,L_i}+L_i p_i^{\,L_i+1}\right)}{(1-p_i)^2}\right)+ \frac{p_i\left(1-(L_i+1)p_i^{\,L_i}+L_i p_i^{\,L_i+1}\right)}{(1-p_i)^2} \quad , \tag{49}$$

$$R_4^i = W_{i0}^{\,2}\left(\frac{1-(4p_i)^{m_i+1}}{1-4p_i}+4^{m_i}\,\frac{p_i^{\,m_i+1}-p_i^{\,L_i+1}}{1-p_i}\right)- 3W_{i0}\left(\frac{1-(2p_i)^{m_i+1}}{1-2p_i}+2^{m_i}\,\frac{p_i^{\,m_i+1}-p_i^{\,L_i+1}}{1-p_i}\right)+2\frac{1-p_i^{\,L_i+1}}{1-p_i} \quad , \tag{50}$$

$$R_5^i = W_{i0}^{\,2}\left(\frac{1-(4p_i)^{m_i+1}}{1-4p_i}-\frac{1-(2p_i)^{m_i+1}}{1-2p_i}\right) - W_{i0}\left(\frac{1-(2p_i)^{m_i+1}}{1-2p_i}-\frac{1-p_i^{\,m_i+1}}{1-p_i}\right)+ W_{i0}\left(2^{m_i}W_{i0}-1\right)\left(2^{m_i}\,\frac{p_i\left(p_i^{\,m_i}-(L_i-m_i+1)p_i^{\,L_i}+(L_i-m_i)p_i^{\,L_i+1}\right)}{(1-p_i)^2}+ \left(2^{m_i}-1\right)\frac{p_i^{\,m_i+1}-p_i^{\,L_i+1}}{1-p_i}\right) \quad . \tag{51}$$

ACKNOWLEDGMENT

We would like to thank Bjørn Selvig for help with the development of the simulation tool used for the validations.

REFERENCES

[1] IEEE 802.11 WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification", IEEE 1999.

[2] IEEE 802.11 WG, "Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)", IEEE 802.11e/D13.0, Jan. 2005.

[3] Bianchi, G., "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", IEEE J-SAC Vol. 18 N. 3, Mar. 2000, pp. 535-547.

[4] Ziouva, E. and Antonakopoulos, T., "CSMA/CA performance under high traffic conditions: throughput and delay analysis", Computer Communications, vol. 25, pp. 313-321, Feb. 2002.

[5] Xiao, Y., "Performance analysis of IEEE 802.11e EDCF under saturation conditions", Proceedings of ICC, Paris, France, June 2004.

[6] Malone, D.W., Duffy, K. and Leith, D.J., "Modelling the 802.11 Distributed Coordination Function with Heterogeneous Load", Proceedings of Rawnet 2005, Riva Del Garda, Italy, April 2005.

[7] Barkowski, Y., Biaz, S. and Agrawal P., "Towards the Performance Analysis of IEEE 802.11 in multihop ad hoc networks", Proceedings of MobiCom 2004, Philadelphia, PA, USA, Sept.-Oct. 2004.

[8] Engelstad, P.E., Østerbø O.N., "Differentiation of the Downlink 802.11e Traffic in the Virtual Collision Handler", Proceedings of the Fifth International IEEE Workshop on Wireless Local Networks (WLN '05), Sydney, Australia, Nov. 15-17, 2005.

[9] IEEE 802.11b WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification: High-speed Physical Layer Extension in the 2.4 GHz Band, Supplement to IEEE 802.11 Standard", IEEE, Sep. 1999.

[10] Kleinrock, L., "Queuing Systems,Vol. 1", John Wiley, 1975.

[11] Wietholter, S. and Hoene, C., "Design and verification of an IEEE 802.11e EDCF simulation model in ns-2.26", Technische Universität Berlin, Tech. Rep. TKN-03-019, November 2003.

[12] Engelstad, P.E., Østerbø O.N., "The Delay Distribution of IEEE 802.11e EDCA ", (Currently under review). The paper will be available at http://www.unik.no/~paalee/research.htm.

Fair power and transmission rate control in wireless networks

Eitan Altman
Maestro project, INRIA
Sophia-Antipolis, France
Email: eitan.altman@inria.fr

Jérôme Galtier
France Telecom R&D
Sophia-Antipolis, France
Email: jerome.galtier@francetelecom.fr

Corinne Touati
Computer Science Department
University of Tsukuba, Japan
Email: corinne@osdp.cs.tsukuba.ac.jp

***Abstract*— In third generation mobile networks, transmission rates can be assigned to both real time and non real time applications. We address in this paper the question of how to allocate transmission rates in a manner that is both optimal and fair. As optimality criterion we use the Pareto optimality notion, and as fairness criterion we use a general concept of which the max-min fairness (which is the standardized fairness concept in ATM networks) and the proportional fairness (which characterizes fairness obtained by some transport protocols for the Internet) are special cases. We show that the problem is a joint optimization system of the transmission rate and the power. We formulate the fair allocation problem as an optimization problem and propose both exact as well as approximating solutions. We consider both uplink and downlink problems and study also macrodiversity.**

I. Introduction

Rate control of calls is an important network management issue in third generation mobile networks. Indeed, not only data transfer but also real time audio and video applications can be transmitted at various rates by selecting an appropriate Codec. In the case of voice applications, UMTS will use the Adaptive Multi-Rate (AMR) codec that offers eight different transmission rates of voice that vary between 4.75 kbps to 12.2 kbps, and that can be dynamically changed every 20 msec. Of course the transmission rate has an impact on the perceived quality. The reduction of the transmission rate is necessary for maintaining a call whose received energy per bit is too small, and it allows to maintain a larger number of calls in the system; it will be studied here in conjunction with power control which is yet another tool that can be used to increase the received energy per bit (but which also has an impact on the interference experienced by other calls).

A well studied problem is that of choosing transmission rates so as to maximize the system's throughput, see [1]. Alternatively, if two mobiles A and B transmitting at the same rate have the same received power at their base station and if A has larger attenuation than B, then it transmits with larger power than B, thus causing more interference than B in the base stations of neighboring cells. Hence systemwise, it is profitable to assign to mobile A lower throughput than to mobile B if there are not sufficient radio resources to assign the maximum throughput to both. This suggests large differences in throughputs assigned to mobiles according to their attenuation level when seeking the system optimal viewpoint.

Yet, a second important consideration in assigning throughputs in networks is fairness. Several fairness concepts have been suggested and implemented in various network architectures, but let us first recall the concept of global optimization.

Global optimization Consider a system with N transmitting source. Let $\underline{r} = (r(1), ..., r(N))$ be a assignment vector of transmission rates. The global optimization maximizes the total throughput $\sum_{n \in N} r(n)$. It can lead to situations in which the allocation is null for one or several sources, and is therefore not consider as fair.

Max-min fairness In ATM networks, the standardized fairness concept in traffic whose rate is controlled (the ABR - Available Bit Rate class) is the so called "max-min fairness" [2]. An assignment vector of rates is said to be max-min fair if one cannot increase the assignment of a source i without decreasing the assignment of a source j for which $r(j) \leq r(i)$ [2], [3]. The quantity that is assigned fairly in ATM is the excess of the throughput beyond a prenegociated minimum transmission rate. Another related standardized fairness concept in ATM is the "weighted max-min fairness" in which the quantity that is to be assigned fairly is the excess throughputs (beyond the minimum guaranteed) weighted by some multiplicative constant depending on the connections.

Proportional fairness In the Internet, the large majority of transfers use the TCP/IP transport control mechanism. The assignment of throughput to various connections using TCP/IP (and other related protocols) can be described using the concept of "proportional fairness" as was shown in [4]. An assignment r is said to be proportionally fair if it is feasible (satisfies the system's constraints) and if for any other feasible assignment $\underline{r}^*$, the aggregate of proportional changes is non-positive:

$$\sum_{i=1}^{N} \frac{r^*(i) - r(i)}{r(i)} \leq 0. \tag{1}$$

The proportional fairness is known to maximize the quantity $\prod_i r(i)$. Equivalently, it is an assignment that maximizes $\sum_i \log r(i)$. A (weighted version of the) proportional fairness is also advocated for future developments of TCP, see e.g. [5]. The way TCP shares bandwidth between connection has become a reference for other real-time applications over the Internet that do not use TCP; such applications are called "TCP friendly".

Both proportional fairness and max-min fairness possess optimality properties: they are both Pareto optimal[1]. The proportional fairness is a good compromise between the system global optimum (i.e. the sum of all mobiles' rates) and the welfare maximization approach of max-min fairness.

Generalized fairness criterion It has been shown in [6] that all three approaches: the system optimization, the max-min fairness and the proportional fairness are all special cases of a generalized fairness concept. Given a positive constant $\alpha \neq 1$, consider the optimization problem:

$$\text{Maximize } \sum_i \frac{r(i)^{1-\alpha}}{1-\alpha}$$

subject to the problem's constraints. Assume that the $r(i)$ are defined on a convex set. Then since the objective function is concave and the constraints are linear, this defines a unique allocation which is called the α-fair allocation. It turns out [6] that this allocation corresponds to the globally optimal allocation when $\alpha \to 0$, to the *proportional fairness* when $\alpha \to 1$, to the *harmonic mean fairness* (another well known fairness concept) when $\alpha \to 2$, and to the *max-min* allocation when $\alpha \to \infty$.

We should mention that other aspects of fair resource assignment in wireless networks have been studied previously. These were aspects related to scheduling back logged packets [7]–[9], so as to achieve already given average transmission rates of different sources. Our study aims, in contrast, to fairly assign the transmission rates.

Our **main contributions** are

1. A methodology for combined rate and power control for achieving arbitrary tradeoff between fairness and system's global maximum for both downlink and uplink as well as for macrodiversity.
2. Transforming a non convex optimization problem into a convex problem with linear constraints. This extends our results in [10] in which only the uplink was considered and which proposed only an approximation in order to derive a convex optimization problem.

The structure of the paper is as follows. We first mention in the next section some other works on rate control in wireless networks (not directly related to fairness concepts). We then introduce in Section III our model and show how the fair assignment problem can be formulated as one of two possible optimization problems: one in which transmission rate can be assigned any real value within some given interval, and one in which finitely many transmission rates are available for each mobile. We analyze the properties of the system in Section IV. We then apply our results to the uplink problem in Section V and propose exact and approximating solutions. The downlink problem is examined in Section VI and the macrodiversity is studied in Section VII. Numerical results are given in Section VIII. A concluding section ends this paper with extensions to utility-based fairness concepts.

[1]An assignment $\underline{r}$ is Pareto optimal if one cannot increase the assignment of one source i without strictly decreasing an assignment of another source j

II. Related work on rate control in wireless networks

We briefly mention in this section some recent papers on rate control in CDMA wireless systems.

The paper that mostly relates to ours is [1]. It considers the problem of optimizing transmission rates and powers. Discrete available rates. The problem is formulated mathematically as a mixed linear-integer programming, for which polynomial optimization algorithms are not available. A heuristic approximation approach based on a Lagrangian is proposed and tested.

Another related paper is [11], where the authors study the optimal control of both the power, and spreading gain (the latter is equivalent to controlling the throughput). The authors restrict to a single cell and to the uplink. The model includes channel coding (FEC, Forward Error Correction), and a general function for BER (Bit Error Rate) as a function of the SIR (Signal to Interference Ratio). Similar models are studied in [12], [13].

Several papers study optimizing throughput assignment for non-real time traffic.

In [14], the quantity that is maximized is the effective data rate: the transmission rate times the BER , where the latter is a function of the SIR and the transmission rate. A Lagrangian approach is used. There is no a priori target SIR. This approach is useful for NRT (Non Real Time) traffic. Note that if we assign to a source a transmission of rate R with 0 percent losses, the effective throughput is the same as if we transmitted at $2R$ and lost half the information. In contrast to NRT, for RT (Real Time) applications, the two scenarios would give different utility (quality perception). In [14] the function of the SIR is taken in an example to be the Shannon capacity; note that approaching that capacity requires long blocks of codes which makes the scheme not useful for RT. There is also a part that considers both RT and NRT (sec VI) but only the throughput of NRT is optimized.

In [15], the authors consider uplink CDMA with two classes, mobiles of the first class (RT) transmit all the time, the other mobiles (NRT) are time-shared. The benefits of time sharing is studied as well as the conditions for silencing some, where one of the studied objectives is that of maximizing throughputs (while keeping the SIR at acceptable levels). The paper takes into account that when a mobile is silent, it still requires energy (for synchronization). We also note that the amount of information to transmit is not changed by the scheduling.

In [16], the author studies the Erlang capacity as a function of the throughputs assigned to NRT applications. Unlike the framework in [1], which we adopt in this paper as well for RT applications, the volume of information transmitted by NRT applications (such as a file transfer) is not affected by the assigned throughput. Thus a static optimization problem, as done in previously mentioned papers, is not adequate to describe the effect of throughput assignment for NRT applications. The model in [16] takes into account the impact of throughput assignment on the call duration in order to

compute the Erlang capacity as a function of the assigned throughput.

Another related research direction has been the assignment of instantaneous rate of packet transmission at the buffers in CDMA wireless systems. In these papers, the actual transmission rate of the source is not controlled.

In [17] the packet transmission rates at the link layer buffers is allocated at as a function of the traffic profiles and is computed according to required bounds on packet losses at these buffers. The paper uses effective bandwidth notions for CBR/VBR traffic (Constant/Variable Bit Rate), and others for ABR (Available Bit Rate). Some other closely related papers are [18]–[20].

In [21], the author considers combined power and rate control of each of a number of queue so as to minimize power and delay. In the considered model, the power and rate assignments determine loss probabilities and there is a given constraint on the loss rate. No retransmissions and no scheduling are considered.

In [22], the authors describe the feasible set of powers/rates in a single cell. They show that this is not a convex set. Convexification is possible by an appropriate time sharing or scheduling of packets. The results are used for assigning transmission rates at the buffers (again, the source rates are given and not controlled), so as to achieve required bounds on delays. The scheduling decisions are taken according to the traffic profile of each mobile which is characterized by the average rate and the burstiness (the so-called $\sigma-\rho$ constraints).

Among all the research directions we mentioned above, our paper is related to the first two references as it is concerned with the actual assignment of transmission throughput rates at the sources (rather than inside the network) of real time applications. We consider a multicell environment with particular attention on uplink and downlink rate control, and include a study on macrodiversity. (This is in contrast to [11] who considers only the uplink control in a single link, or [1] whose framework seems more adapted to the uplink multicell case.) Yet, an important feature of our paper is the introduction of new fairness considerations into the rate allocation problem.

III. The model

We use the notations of [1] applicable for both the uplink and the downlink and extend their model. Consider a cellular radio system with S transmitting sources. Source s can transmit with total power $p_{s,tot}$ within the interval $[0, \overline{p_{s,tot}}]$. In the following, if source s has different channels, $p_{s,t}$ will denote the power of the signal emitted by source s to destination t. Also, $p_{NC,s}$ represents the power of the non-power controlled channels. Therefore:

$$p_{s,tot} = \sum_t p_{s,t} + p_{NC,s}. \tag{2}$$

We take the following notations:

m	a mobile unit
b	a base station
	$(m,b) = \begin{cases} (s,t) & \text{in the uplink case,} \\ (t,s) & \text{in the downlink case.} \end{cases}$
c_m	the cell of mobile m
M	the number of mobile units
N_c	the number of mobiles in cell c
$N(m,b)$	the background noise power at the receiver (it represents thermal noise and also radio interference from non-power controlled channels.)
C	a multiplicative constant
$p'_{m,b}$	the normalized power: $\exists K_{s,t},\ p'_{m,b} = K_{s,t}p_{s,t}$. We will show that : $K_{s,t} = \begin{cases} g_{m,b} & \text{(the link gain) in the uplink,} \\ 1 & \text{in the downlink.} \end{cases}$

Given a power vector $P = (p_{1,tot}, ..., p_{s,tot})$, the received signal to interference ratio of mobile m is given by

$$SIR_m(P) = \frac{p'_{m,b}}{N(m,b) + C \sum\limits_{\substack{m' \neq m \\ m' \in c_m}} p'_{m',b}} \quad 1 \leq m \leq M. \tag{3}$$

As $p'_{s,tot}$ is bounded, then $p'_{m,b}$ is bounded by a value that we will denote by $\overline{p'_{m,b}}$. The values of C, N and p' will be expressed in Section V for the uplink and Section VI for the downlink.

As explained in [1], the above model can be useful for both uplink and downlink. However, we shall later use the particular structure of the uplink and of the downlink in order to simplify the solution.

We next describe two possible settings for the power and transmission rate control.

A. The continuous model

In the first model, mobile m can use any value of throughput between a minimum guaranteed value MR_m and a maximum value PR_m. This can be achieved if a packet mode is used with an appropriate scheduling (see e.g. [23], [24] and references therein). Denote $r(m)$ the transmission rate assigned to mobile m and $R = (r(1), ..., r(M))$ the rate vector. We assume that, for each mobile, there is a minimum required value of SIR_m per transmitted bit per second, which we denote by δ_m.

Let $(E_b/I_o)_i$ be the ratio of bit energy to interference power spectral density of mobile m, and W_m the spreading bandwidth at chip rate for mobile m. We then have

$$\delta_m \leq \frac{1}{W_m} \left(\frac{E_b}{I_o} \right)_m = \frac{SIR_m(P)}{r(m)}, \tag{4}$$

implying $\delta_m r(m) \leq SIR_m(P)$.[2]

[2]Note that we implicitly assume that $(E_b/I_o)_m$ does not depend on the transmission rate $r(m)$. This is a standard assumption in modeling literature, see e.g. [1]. In practice, however, it may depend on $r(m)$, see e.g. [25, p. 151, 222, 239]. But as we see from [25, Fig. 10.4, p. 222], it is close to a constant throughout long range of bit rates. For example, between 16Kbps and 256Kbps, the maximum variation around the median value is less than 20%. We thus propose to take for the value of $(E_b/I_o)_m$ its average or median value over the range $[MR_m, PR_m]$. However, if the exact dependence is available analytically, it can be included into our model.

Thus the solution of our joint power and transmission rate assignment problem is constrained to belong to the set $\Pi^c = (P, R)$, defined through:

$$\begin{cases} 0 \le p_{s,tot} \le \overline{p_{s,tot}}, & 1 \le s \le S, \\ MR_m \le r(m) \le PR_m, & \\ & 1 \le m \le M. \\ \delta_m r(m) \le \dfrac{p'_{m,b}}{N(m,b) + C \sum\limits_{m' \ne m} p'_{m',b}}, & \end{cases} \quad (5)$$

A fair allocation can now be obtained using the following optimization problem:

$$\text{Find } (P, R) \in \Pi^c \text{ that Maximizes } \sum_{m=1}^{M} \frac{r(m)^{1-\alpha}}{1-\alpha}.$$

B. The discrete model

There is a finite number of available transmission rates for each mobile. Let $r_m^1 < r_m^2 < ... < r_m^{K(m)}$ be the available transmission rates for mobile m.

One way to formulate the discrete model is to use the continuous model and add a constraint on the discrete values that the throughputs can have:

$$\text{Find } (P, R) \in \Pi^c \text{ that Maximizes } \sum_{m=1}^{M} \frac{r(m)^{1-\alpha}}{1-\alpha}, \quad (6)$$

$$\text{with } r(m) \in \{r_m^1, r_m^2, ..., r_m^{K(m)}\}. \quad (7)$$

We present below an alternative formulation of the problem following [1].

To properly receive messages at transmission rate r_m^k with tolerable error probability, mobile m is expected to attain an $SIR_m(P)$ not less than a target γ_m^k.

Let $Y = (y_m^k)$ be a 0-1 matrix such that for every mobile m and rate r_i^k,

$$y_m^k = \begin{cases} 1, & \text{if mobile } m \text{ is transmitting with rate } r_m^k, \\ 0, & \text{otherwise.} \end{cases}$$

Introduce arbitrarily chosen constants A_m^k that represent the transmission power that mobile m needs in order to attain γ_m^k, regardless of the interference power. More precisely, they can be chosen arbitrarily so as to satisfy

$$A_m^k \ge \gamma_m^k \Big(N(m,b) + C \times \sum_{\substack{m' \ne m \\ m' \in c_m}} \overline{p'_{m',b}} \Big)$$

for all m and k (we allow in the definition to have $A_m^k \ge \overline{p'_{m,b}}$). Hence the constants A_m^k satisfy

$$A_m^k \ge \max_P \frac{p'_{m,b} \gamma_m^k}{SIR_m(P)},$$

which is in fact the condition that defines these constants in [1].

Then the solution of our joint power and transmission rate assignment problem is constrained to belong to the set $\Pi^d = (P, Y)$ such that:

$$\begin{cases} 0 \le p_{s,tot} \le \overline{p_{s,tot}}, & s = 1, \dots, S, \\ y_m^k \in \{0,1\}, \quad \sum_{k=1}^{K_i} y_m^k \le 1, & \\ & m = 1, \dots, M. \\ p'_{m,b} + (1 - y_m^k) A_m^k \ge \dfrac{p'_{m,b} \gamma_m^k}{SIR_m(P)}. & \end{cases}$$

The first constraint represents the power constraints. The second states that at most one bit rate can be allocated to a mobile. The third constraint reduces to the constraint on SIR_m when r_m^k is the rate allocated to mobile k. For r_m^k which is not the allocated rate, the inequality constraint is just a consequence of the definition of A_m^k.

A fair allocation can now be obtained using the following optimization problem:

$$\text{Find } (P, Y) \in \Pi^d \text{ that Maximizes } \sum_{m=1}^{M} \frac{\left(\sum\limits_{k=1}^{K} y_m^k r_m^k \right)^{1-\alpha}}{1-\alpha}. \quad (8)$$

In this section we made explicit the system of equations corresponding to the continuous and the discrete models. In particular, we showed the need for a join allocation of rates R and power P vectors. In the following section, we focus on the first model.

IV. Properties of the system

A. Equivalent problem

We provide here an equivalent formulation of the problem based on a simple change of variables.

Lemma 1: The continuous problem (see section III-A) is equivalent to finding $(P, \mathcal{C})$ in $\hat{\Pi}^c = (P, \mathcal{C})$ that maximizes $Z(\mathcal{C}) := \sum\limits_{m=1}^{M} \frac{1}{1-\alpha} \left(\frac{\rho(m)}{1 - \delta_m \rho(m)} \right)^{1-\alpha}$ where $\hat{\Pi}^c = (P, \mathcal{C})$ is given by:

$$\begin{cases} 0 \le p_{s,tot} \le \overline{p}_{s,tot}, & 1 \le s \le S, \\ \dfrac{MR_m}{1 + \delta_m MR_m} \le \rho(m) \le \dfrac{PR_m}{1 + \delta_m PR_m}, & \\ & 1 \le m \le M. \\ \delta_m \rho(m) \le \dfrac{p'_{m,b}}{N(m,b) + C \sum p'_{m',b}}, & \end{cases} \quad (9)$$

Proof: Let $\mathcal{C}$ be the N-dimensional vector such that $\forall m = 1 \dots M$, $\rho(m) = \dfrac{r(m)}{1 + C \delta_m r(m)}$. We should notice that $\delta_m \rho(m) = 1 - \dfrac{1}{1 + \delta_m r(m)}$ and simply make the change of variables from R to $\mathcal{C}$ in the inequalities (5). ■

Lemma 2: The objective function Z is concave if for any $\mathcal{C}$ in the set of feasible solutions, we have: $\forall m, 1 \le m \le M$, $2\delta_m \rho(m) \le \alpha$.

Proof: Note that the denominator $1-\delta_m\rho(m)$ is nonnegative over the feasible solutions (from the second inequality of system (5)). To determine whether the objective function is concave, we differentiate it twice with respect to $\rho(m)$, $m=1,...,N_c$ and obtain $\frac{\partial^2 Z(\rho)}{\partial\rho(m)^2} = \frac{2\delta_m\rho(m)-\alpha}{(1-\delta_m\rho(m))^3\rho(m)}\left(\frac{\rho(m)}{1-\delta_m\rho(m)}\right)^{-\alpha}$. This is nonpositive for all feasible $\rho(m)$ if $2\delta_m\rho(m)\leq\alpha$. ■

Remark 1: A sufficient condition for the objective function to be concave is $\alpha\geq 2$. This condition can further be weakened. Let

$$w=\max_{m=1,...,N_c}\frac{\delta_m PR_m}{1+\delta_m PR_m}.$$

Then a weaker sufficient condition for the objective function to be concave is that $\alpha\geq 2w$. Quite often w is close to zero (see e.g. discussion before Lemma 1 in [26]).

In the following, we will call acceptable transmission rate vector any vector R (respectively $\mathcal{C}$) that accepts at least one feasible assignment P satisfying the constraints (5)(respectively (9)).

B. Properties of acceptable rate vectors

We should start by noticing that:

Lemma 3: For any feasible $\mathcal{C}$ and for any cell:

$$C\sum_{m'}\delta_{m'}\rho(m')<1.$$

Proof: Let c be a cell and b its associated base station. Consider the last inequality of system (9):

$$\forall m,\delta_m\rho(m)\leq\frac{p'_{m,b}}{N(m,b)+C\sum_{m'}p'_{m',b}}.$$

As $N(m,b)>0$, then $\delta_m\rho(m)<\frac{p'_{m,b}}{C\sum_{m'}p'_{m',b}}$. By summation, $C\sum_{m'}\delta_{m'}\rho(m')<\frac{C\sum_{m'}p'_{m',b}}{C\sum_{m'}p'_{m',b}}=1$. ■

Lemma 4: Consider the last inequality of system (9) when replaced by equality. We obtain: For a given cell c with base station b, $\forall m\in c$,

$$\delta_m\rho(m)=\frac{p'_{m,b}}{N(m,b)+C\sum p'_{m',b}},\tag{10}$$

Then one can prove that this linear system of N_c equations of N_c variables admits one and only one solution PP for any feasible $\mathcal{C}$.

Proof: It is sufficient to prove that the N_c equations are linearly independent. They can be written as: $\forall m,\delta_m\rho_m N(m,b)=p_{m,b}-C\delta_m\rho_m\sum_{m'}p_{m',b}$ It is of the form $AX=Y$ with: $Y=(\delta_i\rho_i N(i,b))_i$, $X=(p_{i,b})_i$ and $A=Id_{N_c}-B$ with Id_{N_c} the identity matrix of size N_c and B the matrix

$$B=\begin{pmatrix} b_1 & \dots & b_1\\ b_2 & \dots & b_2\\ & \dots & \\ b_{N_c} & \dots & b_{N_c}\end{pmatrix}.\tag{11}$$

with $\forall i, b_i=C\delta_i\rho_i$. If U is an eigenvector of A associated to eigenvalue λ, then $AU=\lambda U=U-BU$. Therefore U is an eigenvector of B with eigenvalue $1-\lambda$. But $\text{rank}(B)=1$ and $\text{trace}(B)=\sum_i b_i$. Then A has only two eigenvalues that are 1 and $1-C\sum_i\delta_i\rho_i$. Therefore A is singular ($1-C\sum_i\delta_i\rho_i\neq 0$ by Lemma (3)) and PP exists and is unique. ■

Proposition 1: For any acceptable fixed transmission rate vector R (respectively $\mathcal{C}$), there corresponds a unique minimum (component wise) power $P'^{\min}$ that satisfies the system (5) (respectively (9)). Moreover $P'^{\min}=PP$.

Proof: As the problems (5) and (9) are equivalent, we only prove the proposition in the first case.

We extend the proof of Lemma 1 in [26] which only considers the single cell case. Suppose that there exists a feasible power assignment P'_0 satisfying the constraints (5). We construct a sequence of power assignments P'_i where

$$\left(p'_{m,b}\right)_{i+1}=\delta_m r(m)\left(N(m,b)+C\sum_{m'\neq m}\left(p'_{m',b}\right)_i\right).$$

We have

$$0\leq\left(p'_{m,b}\right)_{i+1}\leq\left(p'_{m,b}\right)_i.$$

Therefore the decreasing sequence converges to an assignment $P'^{\min}$ satisfying $\forall n, P'^{\min}\leq P'_n$ component wise and

$$\forall m, {p'_{m,b}}^{\min}=\delta_m r(m)\left(N(m,b)+C\sum_{m'\neq m}{p'_{m',b}}^{\min}\right).$$

■

Lemma 5: Let $\mathcal{C}$ be a fixed acceptable transmission vector. The set of feasible power assignments satisfies:

$$\begin{cases}0\leq p_{s,tot}\leq\overline{p_{s,tot}}, & 1\leq s\leq S,\\ \delta_m\rho(m)\Big[C\sum_{m'}\delta_{m'}\rho(m')[N(m',b)-N(m,b)] & \\ \qquad +N(m,b)\Big]\leq\Big(1-C\sum_{m'}\delta_{m'}\rho(m')\Big)p'_{m,b}, & \\ & 1\leq m\leq M.\end{cases}\tag{12}$$

Proof: We consider the last inequality of system (9)

$$p'_{m,b}\geq\delta_m\rho(m)\left(N(m,b)+C\sum_{m'}p'_{m',b}\right),\tag{13}$$

We can now consider cell c separately, and reduce the unknown variables to the power and transmission rate within that cell only. We then sum over all mobiles of cell c:

$$\left(\sum_{m'\in c_m}p'_{m',b}\right)\left(1-C\sum_{m'\in c_m}\delta_{m'}\rho(m')\right)\geq\sum_{m'}\delta_{m'}\rho(m')N(m',b).\tag{14}$$

We finally combine this with (13) to obtain the second inequality of (12). ■

From Proposition 1 and Lemma 5, we get:

Lemma 6: For a given acceptable R (respectively $\mathcal{C}$), ${P'}^{\min}$ is given by $\forall m = 1 \dots M$, ${p'_{m,b}}^{\min} = \delta_m \rho(m) \times$

$$\frac{N(m,b) + C \sum_{m'} \delta_{m'} \rho(m') [N(m',b) - N(m,b)]}{1 - C \sum_{m'} \delta_{m'} \rho(m')}. \quad (15)$$

We finally conclude that:

Theorem 1: A rate vector $\mathcal{C}$ is acceptable if and only if it satisfies the two following conditions.

(C1) $\frac{MR_m}{1 + \delta_m MR_m} \leq \rho(m) \leq \frac{PR_m}{1 + \delta_m PR_m}$,

(C2) $\forall m,\ m = 1 \dots M,\ 0 \leq p_{s,tot}^{min} \leq \overline{p_{s,tot}}$, with $p_{s,tot}^{min}$ defined by (2) and (15).

Proof: If $\mathcal{C}$ is feasible, then (C1) is verified. Proposition 1 states that ${P'}^{\min}$ is a solution of the system, so that (C2) is also satisfied.

Equivalently, if condition (C2) is satisfied, then ${P'}^{\min}$ is a solution vector (it satisfies the first and the third inequalities of (9)). Finally, (C1) is the last inequality of (9). ■

In this section, we focused on the continuous model and provided an equivalent system of equations based on a change of variables (Lemma 1). We then expressed a sufficient condition for the objective function to be concave (Lemma 2 and Remark 1). We showed three properties of this system (Proposition 1, Lemma 5 and Lemma 6). In particular, we showed that if a rate vector is acceptable (that is to say if it corresponds to at least one feasible power assignment), then all the corresponding power vectors are greater (component wise) than the feasible power vector ${P'}^{\min}$ given by: ${p'_{m,b}}^{\min} = \delta_m \rho(m) \times$

$$\frac{N(m,b) + C \sum_{m'} \delta_{m'} \rho(m') [N(m',b) - N(m,b)]}{1 - C \sum_{m'} \delta m' \rho(m')}.$$

We finally concluded with a sufficient and necessary condition for a rate vector to be acceptable (Theorem 1).

V. The uplink case

In this section, we apply the results of the previous section to the uplink case.

A. Continuous case

Let us consider a mobile m in cell c. Let $g_{m,b}$ be the link gain between source m (the mobile) and a destination b (the base station). We assume that time intervals are sufficiently short for $g_{m,b}$ to be constant within the interval. $p_{m,b}$ is the power of the signal emitted by mobile m to its corresponding base station. As mobile m emits only one signal, we have: $p_{s,tot} = p_{m,b}$. Finally, ν_b represents the thermal noise at destination. Then, the SIR_m of mobile m can be written:

$$SIR_m = \frac{g_{b,m}\ p_{m,b}}{\nu_b + \sum_{m' \text{in any cell}, m' \neq m} g_{m',b}\ p_{m'}}. \quad (16)$$

We make the following approximating assumption that is frequently used for the uplink case (see e.g. [27]):

Hypothesis 1: The interference caused by mobiles from other cells is proportional to the interference due to the mobiles in cell c, i.e. there is a constant λ such that $I_{other} = \lambda I_{own}$. In other words:

$$\sum_{m' \text{ in any other cell}} g_{m',b}\ p_{m'} = \lambda \sum_{m' \text{ in cell } c} g_{m',b}\ p_{m'}. \quad (17)$$

Under hypothesis 1, the uplink can therefore be modeled with system (5) (equivalently with system (9)) with:

$$C = \lambda + 1, \qquad N(m,b) = \nu_b, \qquad p'_{m,b} = g_{m,b} p_{m,b}.$$

We then can apply the results of Section IV. In particular, we obtain (see also [16], [27]): $\forall m \in [1, M]$:

$$p_{m,b}^{\min}(\rho) = \frac{1}{g_{b,m}} \left(\frac{\nu_b \delta_m \rho(m)}{1 - (1+\lambda) \sum_{m'=1}^{N_c} \delta_{m'} \rho(m')} \right). \quad (18)$$

Moreover, conditions (C2) of Theorem 1 is now:

$$0 \leq \nu_b \delta_m \rho(m) \leq g_{b,m} \overline{p_{m,b}} \Big(1 - (1+\lambda) \sum_{m' \in c_m} \delta_{m'} \rho(m') \Big).$$

Hence, the problem can then be summarized by: Find $\mathcal{C}$ that maximizes

$$Z(\mathcal{C}) := \sum_{m=1}^{M} \frac{1}{1-\alpha} \left(\frac{\rho(m)}{1 - \delta_m \rho(m')} \right)^{1-\alpha} \text{ s.t.}$$

$$\begin{cases} \frac{MR_m}{1 + \delta_m MR_m} \leq \rho(m) \leq \frac{PR_m}{1 + \delta_m PR_m}, \\ 0 \leq \nu_b \delta_m \rho(m) \leq g_{b,m} \overline{p_{m,b}} \left(1 - (1+\lambda) \sum_{m' \in c_m} \delta_{m'} \rho(m') \right). \end{cases} \quad (19)$$

Remark 2: The set of constraints is now a (convex) polytope.

Remark 3: A sufficient condition for all rate vectors to satisfy Lemma 3 is :

$$\forall c, \qquad (1+\lambda) \sum_{m' \in c} \frac{\delta_{m'} PR_{m'}}{1 + \delta_{m'} PR_{m'}} < 1.$$

We conclude that for $\alpha \geq 2w$ (and in particular for $\alpha \geq 2$, see Lemma 2 and Remark 1), the multicell problem of controlling jointly the power and the transmission rate can be reduced to a standard minimization problem with linear constraints and concave objective function that can be easily solved by either decentralized Lagrangian algorithms or efficient centralized methods based on SDP (Semi Definite Programming), see e.g. [28].

We finally note that for the single cell case, the above solution is an exact one.

B. Further approximations for the uplink solution

The condition $\alpha \geq 2w$ does not cover the interesting case of $\alpha = 0$ which corresponds to the problem of maximizing the global throughput. We therefore propose below two approximations, both applicable for all $\alpha \geq 0$.

1) First approximation scheme: approximating the objective function.

One can approximate the objective function $Z(\rho)$ by $\sum_{m=1}^{M} \frac{(\rho(m))^{1-\alpha}}{1-\alpha}$, i.e. neglect the term $\delta_m \rho(m)$ in the denominator, as it is quite often much smaller than 1 (as mentioned before). With this new objective function replacing the previous one, we obtain a convex optimization problem for any $\alpha > 0$ (except $\alpha = 1$). We note that the constraints (and thus the set of feasible solutions) for this approximating method are the same as in the initial problem. We also note that the value obtained from this approximation is a lower bound for the original optimization problem.

2) Second approximation scheme: approximating the constrained set.

An alternative approximation can be obtained by considering the original formulation (5) in terms of the rate vector R rather than $\mathcal{C}$, in which the objective function is already concave but the constraint set is not convex (see more details on this set in [22] that considers the single cell case). Our approximation then consists in replacing the last constraint in (5) by:

$$\delta_m r(m) \leq \frac{g_{b,m} p_{m,b}}{\nu_b + \sum_{m' \in c_m} g_{b,m'} p_{m'}}, \quad m = 1, ..., M. \quad (20)$$

We can now proceed as in Prop. 1 and consider the equality constraint instead of the inequality, which provides the minimal solution of (20), given by

$$p_{m,b}^{\min}(r) = \frac{1}{g_{b,m}} \left(\frac{\nu_b \delta_m r(m)}{1 - (1+\lambda) \sum_{m' \in c_m} \delta_{m'} r(m')} \right) \quad (21)$$

(see the derivation of Eq. (18).

Substituting this into our new approximation problem, we obtain the equivalent optimization problem of maximizing $\sum_m \frac{r(m)^{1-\alpha}}{1-\alpha}$ over the set Π_{app} of vector R satisfying

$$\Pi_{\text{app}} \begin{cases} MR_m \leq r(m) \leq PR_m, \\ 0 \leq \nu_b \delta_m r(m) \\ \qquad \leq g_{m,b} \overline{p_{m,b}} \Big(1 - (1+\lambda) \sum_{m' in c_m} \delta_{m'} r(m') \Big). \end{cases}$$

We see that the set of constraints is now a (convex) polytope.

Furthermore, let us consider a couple $(p^{\min}, R)$, where $p^{\min}$ is computed in (21) for that R. If it is finite then the couple satisfies the third constraint in original constraint set Π^c. Therefore we replaced the set of constraints by a strict subset of that set. We conclude that the approximating problem gives in fact a *lower bound* on the throughput assignment for each m and a lower bound for the objective function.

C. The discrete model

We finally briefly comment on the discrete model. The solution of the model (8) can be found in the same way as in [1], using a distributed algorithm based on Lagrangian relaxation. Alternatively, one can use the formulation (6). Its solution can follow a similar path as we had for the continuous case: first express for given transmission rates the corresponding minimum power that satisfies the constraint (4). Again the approximation (17) can be used to obtain explicit expressions for optimal power assignments for given transmission rates. This reduces to the same optimization problems we had before, with the same linear constraints, along with the new extra integrity constraint (7).

VI. Downlink solution

Following [29], we write a more precise expression for the signal to interference ratio (Equation (3)) that mobile m connected to base station b experiences: $SIR_{m,b} = \frac{P_{b,m} h_{b,m}}{\nu_m + I_{inter} + I_{intra}}$, with I_{inter} and I_{intra} denoting respectively the intercell and the intracell interference at mobile m. We have :

$$\begin{cases} I_{intra} = \beta (P_{tot,b} - P_{b,m}) h_{b,m} + (1-\beta) P_{SCH,b} h_{b,m}, \\ I_{inter} = \sum_{b'=1, b' \neq b}^{B} P_{tot,b'} h_{b',m}. \end{cases}$$

Equivalently $SIR_{m,b} = P_{b,m} /$

$$\beta \sum_{m' \neq m} p_{b,m'} + P_{sch} + \beta P_{cch} + \frac{1}{h_{b,m}} \Big[\nu_m + \sum_{b'=1, b' \neq b}^{B} P_{tot,b'} h_{b',m} \Big]. \quad (22)$$

where we denote by:

- $P_{b,m}$ the transmission power of base station b to the Dedicated Physical Channel (DPCH) of mobile m,
- $P_{SCH,b}$ the power of the (non orthogonal) synchronization channel from base station b,
- $P_{CCH,b}$ the power of the (orthogonal) common channel from base station b,
- $P_{tot,b}$ the total output power from base station b, given by

$$P_{tot,b} = \sum_{m'=1}^{N_c} P_{b,m'} + P_{CCH,b} + P_{SCH,b}. \quad (23)$$

- $h_{b,m}$ the path gain from base station b to mobile m,
- ν_m receiver's m noise.
- β the synchronization factor,
- B the number of base stations.

Let us denote $F_{b,m}$ the ratio between the received intercell and intracell power, defined as

$$F_{b,m} = \frac{I_{intra}}{I_{inter}}.$$

Then $SIR_{m,b} =$

$$\frac{P_{b,m}}{(1+F_{b,m}) \Big(\beta \sum_{m' \neq m} p_{b,m'} + P_{sch,b} + \beta P_{cch,b} \Big) + \frac{\nu_m}{h_{b,m}}}. \quad (24)$$

As in [29], we shall further approximate $F_{b,m}$ by its average value F, and assume that $P_{CCH,b}$ and $P_{SCH,b}$ are the same for all base stations (b is then omitted). They are known

parameters and are not subject to power control. Also, ν_m does not depend on m.

Then, the downlink joint transmission rates are now determined as the solution of problem (5) (or equivalently problem (9)) with:

$$N_D(b,m) = (1+F)(P_{sch} + \beta P_{cch}) + \frac{\nu}{h_{b,m}},$$

$$C_D = (1+F)\beta \qquad \text{and} \qquad p'_{m,b} = p_{b,m}.$$

We thus obtain this optimization problem: Find $\mathcal{C}$ such that

$$\begin{cases} \dfrac{MR_m}{1+\delta_m MR_m} \le \rho(m) \le \dfrac{PR_m}{1+\delta_m PR_m}, \\ 0 \le p_{SCH} + p_{CCH} + \displaystyle\sum_{m \text{ in cell } c} \delta_m \rho(m), \\ \overline{p_b^{tot}} \ge \\ \dfrac{N_D(m,b) + C_D \displaystyle\sum_{m'} \delta_{m'} \rho(m') \big[N_D(m',b) - N_D(m,b)\big]}{1 - C_D \sum_{m'} \delta_{m'} \rho(m')}. \end{cases} \tag{25}$$

We can notice that for any values of δ, ρ, N and C_D we obviously have:

$$\begin{aligned} &\textstyle\sum_m \delta_m \rho(m) \Big(N_D(m,b) + C_D \sum_{m'} \delta_{m'} \rho(m') \\ &\big[N_D(m',b) - N_D(m,b)\big] \Big) = \textstyle\sum_m \delta_m \rho(m) N_D(m,b). \end{aligned} \tag{26}$$

Moreover $1 - C_D \sum_{m'} \delta_{m'} \rho(m') > 0$ (Lemma (3)). Then: $-(p_{SCH} + p_{CCH})(1 - C_D \displaystyle\sum_{m'} \delta_{m'} \rho(m')) \le 0 \le$ $\displaystyle\sum_m \delta_m \rho(m) N_D(m,b)$.

The optimization problem is finally: Find $\mathcal{C}$ that maximizes $Z(\mathcal{C}) := \displaystyle\sum_{m=1}^{M} \frac{1}{1-\alpha} \left(\frac{\rho(m)}{1 - \delta_m \rho(m')} \right)^{1-\alpha}$ s.t.

$$\begin{cases} \dfrac{MR_m}{1+\delta_m MR_m} \le \rho(m) \le \dfrac{PR_m}{1+\delta_m PR_m}, \\ \displaystyle\sum_{m \text{ in cell } c} \delta_m \rho(m) N_D(m,b) \le \overline{p_b^{tot}} \left(1 - C_D \sum_{m'} \delta_{m'} \rho(m') \right). \end{cases} \tag{27}$$

Then, once again we obtain a minimization problem with linear constraints. For $\alpha \ge 2w$ the objective function is concave and therefore the general problem is convex and solvable in polynomial time.

VII. Macro-diversity in downlink

Many UMTS systems use the possibility for a mobile to receive the signal from several stations. This is called macrodiversity. This prevents the signal from a base station from fading abruptly and as a consequence gives to the mobile a better quality of service.

For the power control part, we shall follow [29]. We consider below soft handover with two base stations or sectors l and s; mobile k has an active link to both stations. We assume maximum ratio combining where the sum of signal to interference ratio should add up to the target value $\delta_k r(k)$:

$$\delta_k r(k) = SIR_{k,l} + SIR_{k,s}. \tag{28}$$

Assume that the link to station s has better signal to interference ratio. Denote

$$\Delta_k = \frac{SIR_{worst\ link}}{SIR_{best\ link}} = \frac{SIR_{k,l}}{SIR_{k,s}} \le 1.$$

In order to solve the joint power and transmission-rate control, we proceed as in [29]. We make the simplifying assumption that Δ_k does not depend on k (we can take the average value among mobiles that are in soft handover).

Let I be the set of mobiles in cell s that do not experience handover. For such mobiles, we have $\delta_i r(i) = SIR_{i,s}$ with $SIR_{i,s}$ given by Equation (24).

Let j be a mobile in soft handover. Then, $\delta_j r(j) = (1+\Delta_j) SIR_{\text{best link}} = \frac{\Delta_j}{1+\Delta_j} SIR_{\text{worst link}}$. Again, its SIR is given by Equation (24). Then, we can distinguish two sets of mobiles : J is the set of mobiles which best link is with base station s, and K is the set of mobiles also experiencing soft handover, but which worst link is with base station s.

Equation (23) becomes:

$$P_{tot,b} = \sum_{i \in I} P_{i,b} + \sum_{j \in J} P_{j,b} + \sum_{k \in K} P_{k,b} + P_{CCH,b} + P_{SCH,b}.$$

Define:

$$\begin{cases} N_I(b,m) = N_D(b,m), & C_I = C_D \\ N_J(b,m) = \frac{1}{\Delta+1} N_D(b,m), & C_J = \frac{1}{\Delta+1} C_D \\ N_K(b,m) = \frac{\Delta}{\Delta+1} N_D(b,m), & C_K = \frac{\Delta}{\Delta+1} C_D \end{cases}$$

Note, in contrast that the authors of [29] do not distinguish between the sets J and K, that is why their equation differs. Also, they assume that the number of mobile in all cell is constant, and that $\forall i,j, \delta_i r(i) = \delta_j r(j)$.

Let $i(m)$ be the set that m is belonging to ($i(m) \in \{I,J,K\}$). We get the following optimization problem for determining the transmission rates:

Find $\mathcal{C}$ that maximizes $Z(\mathcal{C}) = \displaystyle\sum_{m=1}^{M} \frac{1}{1-\alpha} \left(\frac{\rho(m)}{1 - \delta_m \rho(m')} \right)^{1-\alpha}$ s.t.

$$\begin{cases} \dfrac{MR_m}{1+\delta_m MR_m} \le \rho(m) \le \dfrac{PR_m}{1+\delta_m PR_m}, \\ \displaystyle\sum_{m \text{ in cell } c} \delta_m \rho(m) N_{i(m)}(b,m) \\ \qquad \le \overline{p_b^{tot}} \left(1 - C_{i(m)} \textstyle\sum_{m'} \delta_{m'} \rho(m')\right). \end{cases} \tag{29}$$

We thus obtain again an optimization problem with concave objective function for all $\alpha \ge 0, \alpha \ne 1$ and linear constraints, which is standard to solve and has efficient solutions.

VIII. Numerical tests

In the following, we show some of the results obtained using the program (19) for the uplink. For M mobiles, we consider a single cell and set $\forall m, \delta_m = 1$, $\nu = 1$, $\overline{p_{m,b}} = 1/M$, $MR_m = 1/4M$, and $PR_m = 1$. The position of the mobiles is taken at random in the $[-1;+1] \times [-1;+1]$ square, and the gain $g_{b,m}$ is equal to $1/d_m^2$, where d_m is the distance of the mobile to the center of the square. We take $M = 50$.

On figures 1 and 2 we show some results obtained for α equal to 0 and 1 corresponding to the global optimization and the proportional fairness respectively. The base station is represented as a black circle in the middle of the figure, and the mobiles are represented with circles centered at their location, and whose radius is proportional to the throughput assigned. Clearly we see that the mobiles closer to the base station tend to receive more bandwidth. Also many subtle differences appear between the two fairness criteria. In particular, proportional fairness allocate no mobile to its maximum throughput demand PR and redistributes the bandwidth to in-between users. Further users still receive the minimum bandwidth MR.

IX. Concluding remarks and extensions

In this paper we addressed the problem of joint transmission rate and power control in wireless networks so as to be both fair and optimal.

A question not addressed here is how to achieve these throughputs in practice if packet mode is used, or in other words, how to schedule packets in order to achieve the throughputs that were fairly assigned. This question has been well studied see e.g. [7]–[9].

The paper is in line with many references [4], [6], [30], [31] that considered the throughput as the object to be fairly assigned, in other networking contexts. In the fairness analysis, one may consider the case in which the utility corresponding to the transmission rates should be fairly assigned, rather than directly the throughputs. Indeed, since utility represents the degree of satisfaction as a function of the assigned throughput, which may be application dependent, assigning the same throughput to two applications might be highly unfair. In fact, mathematical frameworks for defining fairness indeed exist, within the area of cooperative game theory, and they always relate to utilities. The central concept of this type that has been applied to fair resource allocation problems is the so called Nash Bargaining solution [28], [31], [32]; it turns out to agree with the proportional fairness concept when utilities are linear. If $f_j(r(j))$ is the utility for mobile j to have a transmission rate of $r(j)$, then the Nash Bargaining Solution is given by the solution of the optimization problem:

$$(P_J) \quad \max \prod_{i=1}^{M} (f_i(r(i)) - f_i(MR_i)), \tag{30}$$

where the maximization is over the appropriate constrained set (Π^c in the continuous model, Π^{app} for the corresponding approximating problem, and Π^d in the discrete problem), see [28]. For real-time voice applications, f_i are typically concave functions over the interval $[MR_i, PR_i]$, which implies that the objective function in the above problem is concave. In particular, if we consider use the approximating approach to compute bounds for the continuous model and solve P_j over the constrained set Π^{app}, this is then again a standard concave optimization problem with linear constraint for which many efficient (polynomial) methods exist. In particular, several solution methods are proposed in [28] for such problems if we express the utilities using quadratic functions.

Acknowledgment

We wish to thank Mr. Jean-Marc Kelif and Dr. Zwi Altman for their many useful discussions.

References

[1] Z. R. S. L. Kim and J. Zander, "Combined power control and transmission selection in cellular networks," in *Proc. of IEEE Vehicular Technology Conference*, Fall 1999.

[2] "Traffic management specification," The ATM forum Technical Committee, April 1996, version 4.0.

[3] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, 1987.

[4] A. M. F. P. Kelly and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, 1998.

[5] J. Crowcroft and P. Oechslin, "Differentiated end-to-end internet services using a weighted proportional fair sharing tcp," *Computer Communications Review*, vol. 28, no. 3, pp. 53–67, July 1998.

[6] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," in *Proc. of SPIE International Symposium on Voice, Video and Data Communications*, 1998.

[7] S. L. V. Bharghavan and T. Nandagopal, "Fair queuing in wireless networks, issues and applications," *IEEE Personal Communications*, vol. 6, no. 1, Feb 1999.

[8] G. Miklós and S. Molnár, "Fair allocation of elastic traffic for a wireless base station," in *Proc. of IEEE Globecom*, Rio de Janeiro, Dec. 1999, pp. 1673–1678.

[9] V. B. S. Lu and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Trans. on Networking*, vol. 7, no. 4, pp. 473–489, 1999.

[10] C. T. E. Altman, J. Galtier, "Fair power and transmission rate control in wireless networks," in *IEEE Globecom*, Taipei, Taiwan, Nov. 2002.

[11] S. Oh and K. M. Wasserman, "Optimality of greedy power control and variable spreading gain in multiclass cdma mobile networks," in *Proc. of Mobicom*, Seattle, Washington, USA, 1999, pp. 102–112.

[12] ——, "Dynamic spreading gain control in multiservice cdma networks," *IEEE J. Selected Area in Comm.*, vol. 17, no. 5, pp. 918–927, May 1999.

[13] ——, "Adaptive resource allocation in power constrained cdma mobile networks," in *Proc. IEEE WCNC*, Sept. 1999, pp. 510–514.

[14] W. S. W. C. W. Sung, "Power controll and rate management for wireless multimedia cdma systems," *IEEE Trans. on Communications*, vol. 49, no. 7, pp. 1215–1226, Jul 2001.

[15] S. Ramakrishna and J. M. Holtzman, "A scheme for throughput maximization in a dual-class cdma system," *IEEE Journal Selected Areas in Comm.*, vol. 16, pp. 830–844, Aug. 1998.

[16] E. Altman, "Capacity of multi-service cdma cellular networks with best-effort applications," INRIA," Research Report, March 2002.

[17] J. W. M. D. Zhao, X. Shen, "Quality-of-service support by power and rate allocation in mc-cdma systems," in *Proc. IEEE GLOBECOM*, Nov. 2001, pp. 604–608.

[18] J. W. M. M. Cheung, "Resource allocation in wireless networks based on joint packet/call levels qos constraints," in *Proc. of IEEE Globecom*, Nov 2000, pp. 271–275.

[19] S. Z. J. W. Mark, "Power control and rate allocation in multirate wideband cdma systems," in *Proc. IEEE WCNC*, Sept 2000, pp. 168–172.

[20] J. W. M. L. Xu, X. Shen, "Performance analysis of adaptive rate and power control for data service in ds-cdma systems," in *Proc. of IEEE GLOBECOM*, Nov. 2001, pp. 627–631.

[21] R. Berry, "Power and delay trade-offs in fading channels," Ph.D. dissertation, Massachusets Institute of Technology, June 2000.

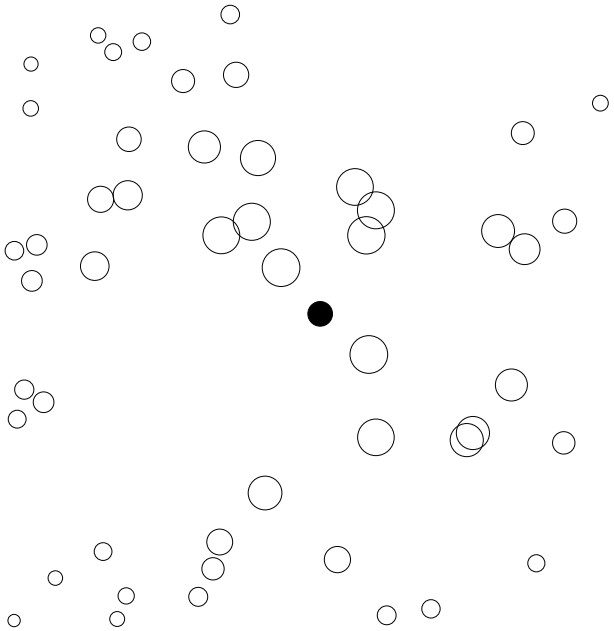


Fig. 1. Throughput distribution with $\alpha = 0$.

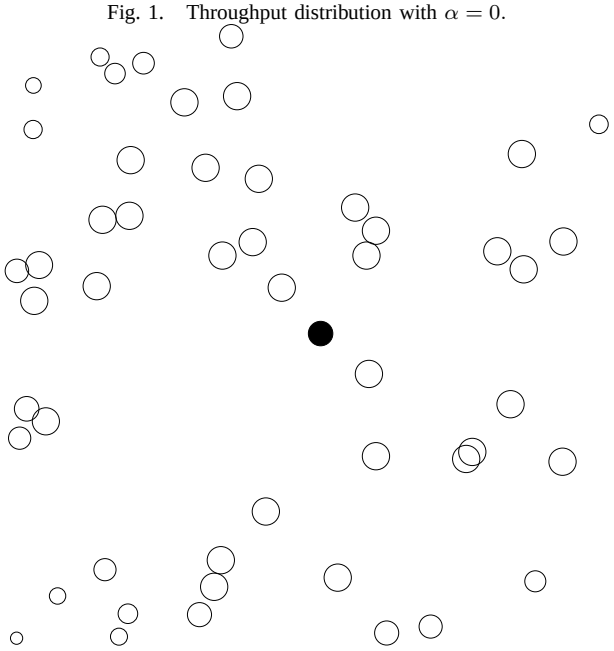


Fig. 2. Throughput distribution with $\alpha = 1$.

[22] R. Leelahakriengkrai and R. Agrawal, "Scheduling in multimedia ds-cdma wireless networks," Technical Report ECE-99-3, July 1999, university of Wisconsin - Madison.

[23] F. C. A. Baiocchi and C. Martello, "Optimizing the radio resource utilization of multiaccess systems with a traffic-transmission quality adaptive packet scheduling," *Computer Networks*, vol. 38, pp. 225–246, 2002.

[24] R. V. M. Ferracioli, V. Tralli, "Channel adaptive scheduling for a wide-band tdd/tcdma wireless system under heterogeneous traffic conditions," *Computer Networks*, vol. 38, pp. 207–223, 2002.

[25] H. Holma and A. Toskala, *WCDMA for UMTS*. J. Wiley & Sons, 2001, revised Edition.

[26] D. Ayyagari and A. Ephremides, "Power control for link quality protection in cellular ds-cdma networks with integrated (packet and circuit) services," in *Proc. of Mobicom'99*, Seattle Washington, USA, 1999, pp. 96–102.

[27] J. Laiho and A. Wacker, "Radio network planning process and methods for wcdma," *Ann. Telecommun.*, vol. 56, no. 5-6, 2001.

[28] C. Touati, E. Altman, and J. Galtier, "On fairness in bandwidth allocation," INRIA Research Report, Sept. 2001.

[29] K. Hiltunen and R. D. Bernardi, "Wcdma downlink capacity estimation," in *Proc. of VTC'2000*, 2000, pp. 992–996.

[30] L. Massouillé and J. W. Roberts, "Bandwidth sharing and admission control for elastic traffic," *Telecommunication Systems*, 2000.

[31] R. R. M. H. Yaiche and C. Rosenberg, "A game theoretic framework for bandwidth allocation and pricing in broadband networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 667–677, 2000.

[32] R. Mazumdar and C. Douligeris, "Fairness in network optimal flow control: Optimality of product forms," *IEEE Trans. on Comm.*, vol. 39, pp. 775–782, May 1991.

A Localized Authentication, Authorization, and Accounting (AAA) Protocol for Mobile Hotspots

Sungmin Baek, Sangheon Pack, Taekyoung Kwon, and Yanghee Choi
School of Computer Science and Engineering
Seoul National University, Seoul, Korea
Email: {smbaek, shpack}@mmlab.snu.ac.kr and {tkkwon, yhchoi}@snu.ac.kr

***Abstract*— Mobile hotspots, i.e. Internet access services in moving networks (e.g. vehicular area networks (VAN) and personal area networks (PAN)) bring about new challenging issues. Even if the network mobility (NEMO) basic support protocol has been standardized as a mobility solution by the Internet Engineering Task Force (IETF), to the best of our knowledge, no studies have been conducted in the area of authentication, authorization, and accounting (AAA) protocol, which is a core technology for public mobile hotspots. In this paper, we propose a localized AAA protocol to retain the mobility transparency as the NEMO basic support protocol and to reduce the cost of the AAA procedure. In addition to providing mutual authentication, the proposed AAA protocol prevents various threats such as replay attack, man in the middle attack, and key exposure. Also, we develop an analytical model to evalutate the AAA signaling cost. Numerical results reveal that the proposed AAA protocol is a suitable solution for AAA services in different mobile hotspots.**

I. Introduction

With the advances of wireless access technologies (e.g., 3G, IEEE 802.11/16/20) and mobile communication services, the demand for Internet access in mobile vehicles such as trains, buses, and ships is constantly increasing [1]. In these vehicles, there are multiple devices constituting a vehicular area network (VAN) or personal area network (PAN) that may access to Internet. This kind of services are referred to *mobile hotspots* [2]. Recently, many studies have been conducted for mobile hotspots [3] [4] [5].

In terms of mobility management, the Internet Engineering Task Force (IETF) has established a working group called *NEMO* [6] and the NEMO working group has proposed an extended Mobile IPv6 protocol [7], i.e. the NEMO basic support protocol [8]. Throughout this paper, we assume the NEMO basic support protocol as a framework.

According to the terminologies in [9], a mobile network (MONET) is defined as a network whose point of attachment to the Internet varies as it moves about. A MONET consists of mobile routers (MRs) and mobile network nodes (MNNs). Each MONET has a home network to which its home address belongs. When the MONET is in the home network, the MONET is identified by its home address (HoA). On the other hand, the MONET configures a care-of-address (CoA) on the egress link when the MONET is away from the home network. At the same time, on the ingress link, the MNNs of the MONET configures CoAs, which are derived from the subnet prefix (i.e. mobile network prefix (MNP)). The MNP remains assigned to the MONET while it is away from the home network. The assigned MNP is registered with the home agent (HA) according to the NEMO basic support protocol [8].

The main goal of the NEMO basic support protocol is to preserve established communications between the MONET and correspondent nodes (CNs) during movements. Packets sent by CNs are first addressed to the home network of the MONET. Then, the HA intercepts the packets and tunnels them to the MR's registered address, i.e. the CoA on the egress link. To deliver packets towards the MR's CoA, the NEMO basic support protocol makes a bi-directional tunnel between the HA and the MR. This tunneling mechanism is similar to the solution proposed for host mobility support, i.e. Mobile IPv6 [7] without route optimization.

To make mobile hotspots feasible in public wireless Internet, well-defined authentication, authorization, and accounting (AAA) protocols should be accompanied. However, to the best of our knowledge, no specific AAA protocols have been proposed for mobile hotspots. Even if a number of AAA protocols have been proposed for host mobility, all of them are based on per-node AAA operations. Therefore, they cannot be directly applied to the MONET containing two different types MNNs: *local fixed nodes (LFNs)* and *visiting mobile nodes (VMNs)*. An LFN belongs to the subnet to the MR and is unable to change its point of attachment, while a VMN is temporarily attached to the MR's subnet by obtaining its CoA from the MNP. The VMN's home network may have different administrative policy (e.g. billing) from the current attached MONET. Therefore, a new AAA procedure for VMNs is required.

In this paper, we propose a localized AAA protocol that provides efficient AAA procedures for both LFNs and VMNs in mobile hotspots.

Our main contributions are summarized as follows.

1) The proposed AAA protocol is consistent with the NEMO basic support protocol. In other words, individual AAA operations for LFNs within a MONET are not performed; instead, the MR is authenticated on behalf of the LFNs. Conversely, each VMN attached to the MONET performs its AAA operation in an individual manner.
2) The proposed AAA protocol localizes the AAA procedure using a local AAA key when the MR hands off within the same foreign network. Therefore, the AAA traffic (also, AAA latency) can be reduced significantly.

3) The proposed AAA protocol allows mutual authentication. In addition, it prevents various security attacks, e.g. replay attack, man in the middle attack.
4) From the point of view of internet service providers (ISPs), how to charge a VMN for its network usage is a critical issue. The proposed AAA protocol supports an flexible billing mechanism in which the VMN is informed of a billing agreement between the MR's home network and the new foreign network. Accordingly, the proposed AAA protocol is a suitable solution when the MONET hands off between different networks with different billing or service policies.

The remainder of this paper is organized as follows. In Section II, an existing AAA protocol for Mobile IPv6 is introduced as a reference protocol. Section III describes the proposed AAA protocol for mobile hotspots. In Section IV, security of the AAA protocol is analyzed. In Section V, an analytical model for the AAA signaling cost is developed and numerical results are presented. Section VI finally concludes this paper.

II. AAA Protocol in Mobile IPv6

In this section, the AAA protocol in Mobile IPv6 is described as a reference model. Although several AAA protocols have been proposed in the literature, we adopt the DIAMETER extension for Mobile IPv6 protocol [11], which is the only valid IETF Internet draft as of this writing. The DIAMETER extension for Mobile IPv6 allows a Mobile IPv6 node to access a network of a service provider after the AAA procedures based on the DIAMETER protocol [10] is completed.

This protocol assumes a network architecture for AAA services, as illustrated in Figure 1. The AAAv is an AAA server in the foreign network, while the AAAh is an AAA server in the home network of the mobile node (MN). The AAA client operates in an entity in a foreign network. Hereafter, we assume that the AAA client is located at each access router (AR). The AAA client performs three tasks: (a) allowing the MN to be authenticated, (b) generating accounting data for the MN's network usage, and (c) authorizing the MN to use network resources.

In addition, followings are assumed by [11].

- An MN is identified by its network access identifier (NAI) [12], which is globally unique.
- An MN and its AAAh have a long-term key.
- Communication between the AAAv and AAAh is secure.

The basic information flow of the DIAMETER extension for Mobile IPv6 [11] is shown in Figure 2. When entering a new network or at power up, an MN listens to an AR's router advertisement (RA) message that has a local challenge and a visited network identifier. Then, the MN sends an authentication request (`AReq`) message to the AAA client (i.e. AR) based on the security key shared with its

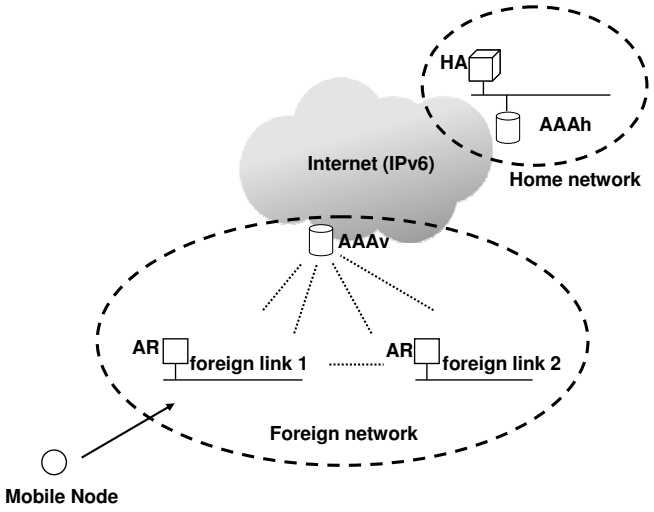


Fig. 1. Mobile IPv6 AAA Architecture

AAAh. When the AAA client receives an `AReq` message, it creates an `AA-Registration-Request Command (ARR)` message and sends it to the AAAv. Then, the AAAv relays it to the AAAh of the MN. When receiving the `ARR` message from the AAAv, the AAAh authenticates the MN by means of the NAI and sends the `Home-Agent-MIPv6-Request Command (HOR)` message to the MN's HA. Upon receipt of the `HOR` message, the HA creates a key to establish a security association (SA) with the MN and replies with the `Home-Agent-MIPv6-Answer Command (HOA)` message to the AAAh. Then, the AAAh constructs the `AA-Registration-Answer Command (ARA)` message that has an authentication result and sends it to the AAAv. When receiving the `ARA` message from the AAAh, the AAAv stores the authentication result locally and then forwards the message to the AAA client. The AAA client converts the `ARA` message into the authentication reply (`ARep`) message in order to inform the MN of the authentication result from the AAAh and deliver the established key (for the SA) to the MN.

III. AAA Protocol for Network Mobility

A. Network Architecture

In this section, the AAA architecture for mobile hotspots is introduced with basic assumptions and concepts such as SA and challenge/response authentication. Figure 3 illustrates the reference AAA architecture in mobile hotspots, which is similar to that of Mobile IPv6. The AAA architecture is based on the DIAMETER protocol [10].

The AAA architecture consists of multiple autonomous wireless networks, each of which is called a *domain*. Each domain has an AAAH server and/or an AAAL server in order to authenticate any node in a DIAMETER-compliant manner. The AAAH server of the MR has the profile of the MR and it shares a long-term key with the MR. Likewise, the AAAH server of the VMN shares a long-term key with the VMN.

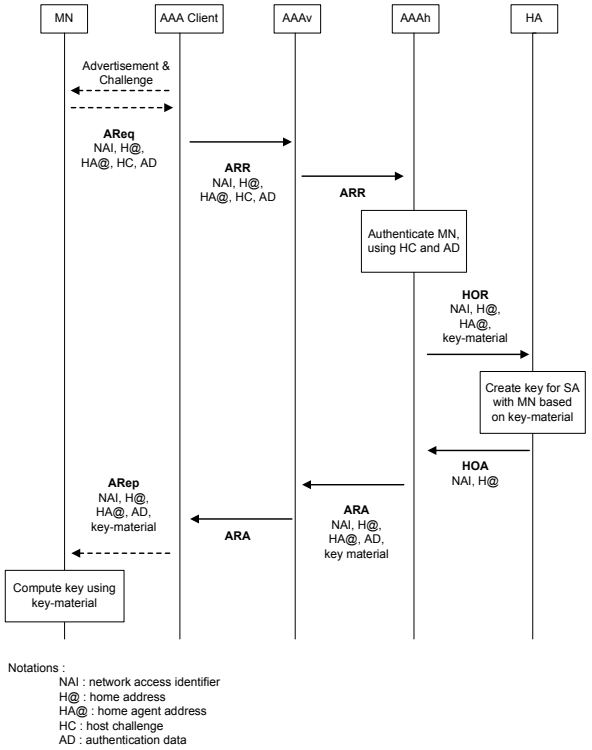


Fig. 2. Information flow in AAA protocol for Mobile IPv6

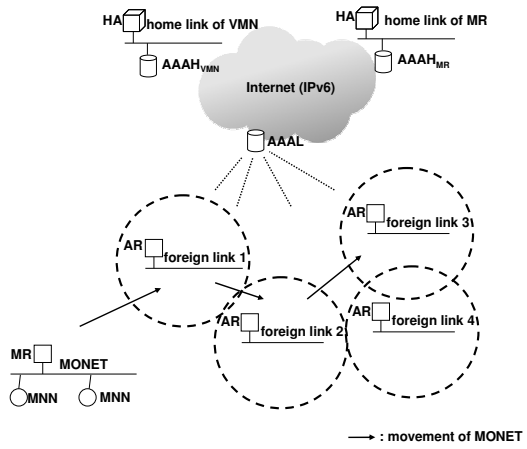


Fig. 3. An AAA Architecture for mobile hotspots

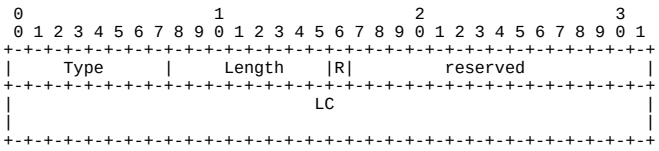


Fig. 4. Attendant advertisement option

The AAAL server takes charge of an AAA procedure for a visiting MONET (i.e. VMNs and MRs). The trust relationship between the MR's AAAH server and the AAAL server in the visited network is maintained through the DIAMETER protocol. When the MONET changes its point of attachment, the MR needs to be authenticated and authorized before it accesses a new domain in the same foreign network (i.e. *intra-domain handoff*) or a new foreign network (i.e. *inter-domain handoff*). To accomplish this, the MR and AR authenticate each other through a mutual authentication procedure that involves both the AAAH server of the MR and the AAAL server of the AR. An attendant (which is the same as a AAA client) is an entity that triggers authentication procedures to the AAA system. In Mobile IPv6 networks, ARs normally act as the attendants for an MN. In our protocol, the AR serves as an attendant for the MR's authentication, whereas the MR serves as an attendant for VMN's authentication. In the latter case, the MR broadcasts attendant advertisement messages and receives authentication request messages from VMNs within the mobile hotspot. In other words, an attendant (an AR or MR) requests the AAAL server to authenticate the mobile hotspot (the MR or VMN). When the AAAL server receives this authentication request, it verifies the identity of the MONET by cooperating with an AAAH server.

In terms of SAs, it is assumed that the MR's AAAH server and the VMN's AAAH server have a pre-established SA. In addition, it is assumed that the MR and LFNs have already authenticated each other by some mechanism, which is out of the scope of this paper.

Notations used throughout this paper are summarized in Table I. A local challenge (*LC*) is a random number for authentication procedures. An MR or VMN encrypts the *LC* using a pre-defined SA with its AAAH server. The encrypted value is called a credential (CR), which is used to authenticate an MR which creates it. MRs and VMNs are identified by their NAIs and a replay protection indicator (RPI) is used to protect from a replay attack. Either a time stamp or a random number can be used as an RPI. The size of the K_{AAA} field is 128 bytes by assuming a public key cryptography algorithm. As we adopt a symmetric key cryptography for dynamic keys (K_{LOCAL} and K_{HOME}, the size of each key is 32 bytes. Note that a dynamic key is used to establish a dynamic SA while a long-term key is to establish a long-term SA. Other notations will be elaborated later.

In our protocol, we define two ICMP messages [18] `Attendant Solicit` and `Attendant Advertisement` messages that are similar to `Router Solicit` and `Router Advertisement` messages, respectively. In those messages, we introduce a new `Attendant Solicit option`, which is used for the authentication of VMNs in case of intra-domain handoff. In addition, several DIAMETER messages, e.g. `AA-Mobile-Router-Request` , `AA-Mobile-Router-Answer`, are defined. Their functions will be described later.

TABLE I

NOTATIONS FOR THE AAA PROTOCOL

Field	Meaning	Typical Length (bytes)
LC	local challenge	8
MC	mobile challenge	8
NAI	identity of MR or VMN	20
RPI	replay protection indicator	4
H@	home address	16
HA@	home agent address	16
Co@	care of address of MR or VMN	16
K_{AAA}	pre-shared SA between an MR and an AAAH server	128 (public key)
K_{AH}	pre-shared SA between an AAAH server and an HA	128 (public key)
K_{AL}	pre-shared SA between an AAAH server and an AAAL server	128 (public key)
CR	credential	8
CR_L	local credential	8
CR_M	mobile credential	8
K_{LOCAL}	dynamic SA between an MR and an AAAL server	32 (symmetric key)
K_{HOME}	dynamic SA between an MR and its AAAH server	32 (symmetric key)
SP_{LOCAL}	security parameters for constructing K_{LOCAL}	12
SP_{HOME}	security parameters for constructing K_{HOME}	12

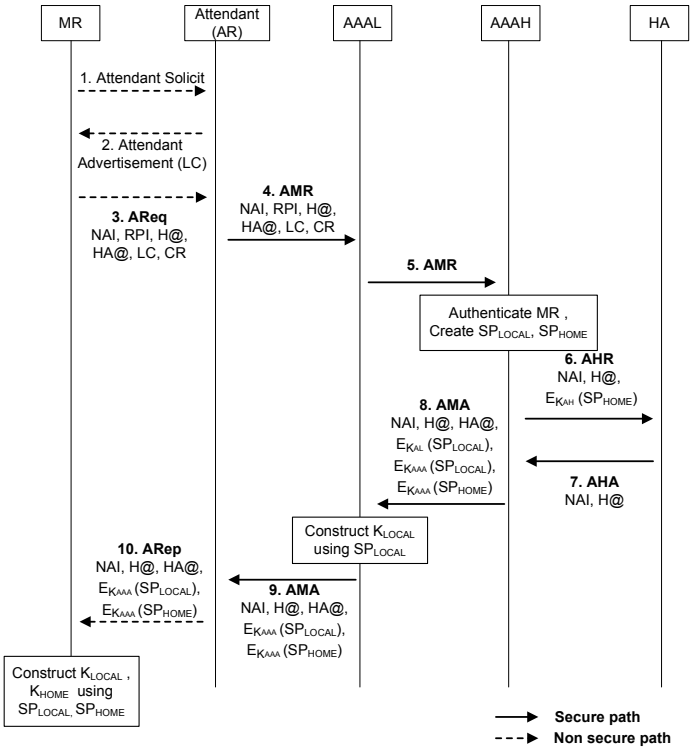


Fig. 5. The AAA procedure of an MR when the inter-domain handoff occurs

B. Mobile Router (MR) Authentication

1) Inter-Domain AAA Procedure: When a MONET enters a new foreign network domain, an inter-domain AAA procedure is initiated. Since the MR does not have any SA with the AAAL server in the foreign network domain, it should be authenticated with its AAAH server located in its home network domain. The message flows for the inter-domain AAA procedure are depicted in Figure 5.

1) The MR sends an `Attendant Solicit` message to the attendant, i.e. AR.

2) As a response to the `Attendant Solicit` message, the AR sends an `Attendant Advertisement` message including an *LC*. Even without the `Attendant Solicit` message, the AR broadcasts `Attendant Advertisement` messages periodically.

3) The MR encrypts the received *LC* value using its long-term SA with the AAAH server and makes a *CR*, which is used for the MR's AAAH server to authenticate the MR. Then, the MR sends an `AReq` message that contains the *LC* and *CR* to the attendant (i.e. AR). The `AReq` message also contains the MR's NAI and RPI, which are used for the AAAL server to identify the MR's home domain and to protect from replay attack.

4) When the attendant (i.e. AR) receives the `AReq` message, the attendant converts it into an `AA-Mobile-Router-Request(AMR)` message. After then, the attendant sends the `AMR` message to the AAAL server in the foreign domain.

5) The AAAL server detects that it cannot authenticate the MR locally by checking the NAI field and hence forwards the `AMR` message to the MR's AAAH server.

   When the AAAH server receives the `AMR` message, it encrypts the *LC* using the pre-established SA and compares the result with the *CR* value. If these two values are identical, the MR is successfully authenticated. Then, the AAAH server generates two dynamic keys: one is a K_{LOCAL} (to be explained later) for intra-domain AAA procedures in the foreign domain and the other is a K_{HOME} for a secure

bi-directional tunnel between the MR and the MR's HA.

To enable the MR to generate K_{LOCAL} and K_{HOME}, the AAAH server also generates SP_{HOME} and SP_{LOCAL} and sends them to the MR. These security parameters are encrypted using the long-term key between the MR and AAAH server to avoid the possibility of exposure to other network entities.

6) The AAAH server informs the HA of the MR's NAI and SP_{HOME} by the `AA-Home-Agent-Request (AHR)` message.

7) The HA constructs K_{HOME} by using SP_{HOME} and replies with an `AA-Home-Agent-Answer (AHA)` message as confirmation.

8) The `AA-Mobile-Router-Answer (AMA)` message is used for the AAAH server to notify the AAAL server of the authentication result. When the AAAL server receives the `AMA` message with authentication approval, the AAAL server decrypts the message using the long-term key (K_{AL}) with the AAAH server, records the MR's NAI, and constructs K_{LOCAL}.

9) The AAAL server re-encrypts the received `AMA` message from the AAAH server after excluding $E_{K_{AL}}(SP_{LOCAL})$ and sends it to the attendant.

10) When receiving the `AMA` message, the attendant learns that the MR is authenticated and grants the MR's network access. In addition, the attendant informs the MR of the result by the `ARep` message containing SP_{HOME}, SP_{LOCAL}, home agent address, etc.
On receipt of the `ARep` message with authentication approval, the MR can access the foreign network. At the same time, the MR generates K_{HOME} and K_{LOCAL} using SP_{HOME} and SP_{LOCAL}, respectively.

2) Intra-Domain AAA Procedure: To support real-time multimedia applications in mobile hotspots, it is important to reduce the latency related to AAA operations. Therefore, when a MONET changes its point of attachment within the same foreign domain, our protocol enables the MR to be authenticated through a localized AAA procedure with the AAAL server in the foreign network without any interaction with its AAAH server. That is, the AAAL server of the foreign network can authenticate the MR using K_{LOCAL}, which was introduced for the inter-domain AAA procedure in the previous section.

Figure 6 illustrates the intra-domain AAA procedure. As a response to the `Attendant Advertisement` message, the MR sends the `AReq` message containing CR_L, which is different from CR used in the inter-domain AAA procedure. At this time, the `AReq` message contains MC for

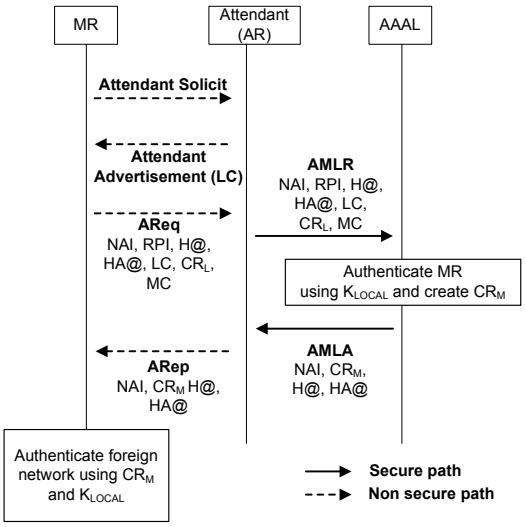


Fig. 6. The AAA procedure of an MR when the inter-domain handoff occurs

mutual authentication. The CR_L is an authentication code generated using K_{LOCAL}. Then, the attendant constructs an `AA-Mobile-Router-Local-Request (AMLR)` DIAMETER message and sends it to the AAAL server. When the AAAL server receives the `AMLR` message, the AAAL server authenticates the MR by using K_{LOCAL}, which has been already stored at the AAAL server during the inter-domain AAA procedures. Moreover, the AAAL server constructs CR_M by encrypting the MC value and informs the attendant of the result via the `AA-Mobile-Router-Local-Answer (AMLA)` message. Then, the attendant will transmit the result (i.e. the `ARep` message) to the MR. The MR receiving the `ARep` message also verifies the CR_M value to authenticate the foreign network.

C. Visiting Mobile Node (VMN) Authentication

A VMN is a visiting MN that accesses the Internet through an MR in mobile hotspot services. According to the NEMO basic support protocol [8], the VMN does not need to know whether its attached router is the AR or the MR. Therefore, the AAA protocol for VMNs should be consistent with this issue. The VMN in a MONET uses the home network prefix of the MR as its IPv6 network prefix. Accordingly, the VMN will deem it to be in the MR's home network. In our AAA protocol, the MR serves as an attendant for VMNs and the MR's AAAH server serves as an AAAL server.

Figure 7 illustrates message flows for the AAA procedure when a VMN is attached to a MONET. As mentioned above, the MR acts as an attendant. Hence, the MR broadcasts `Attendant Advertisement` messages periodically or responds to an `Attendant Solicit` message from the VMN with an `Attendant Advertisement` message. The VMN creates a CR using a pre-shared SA with its AAAH

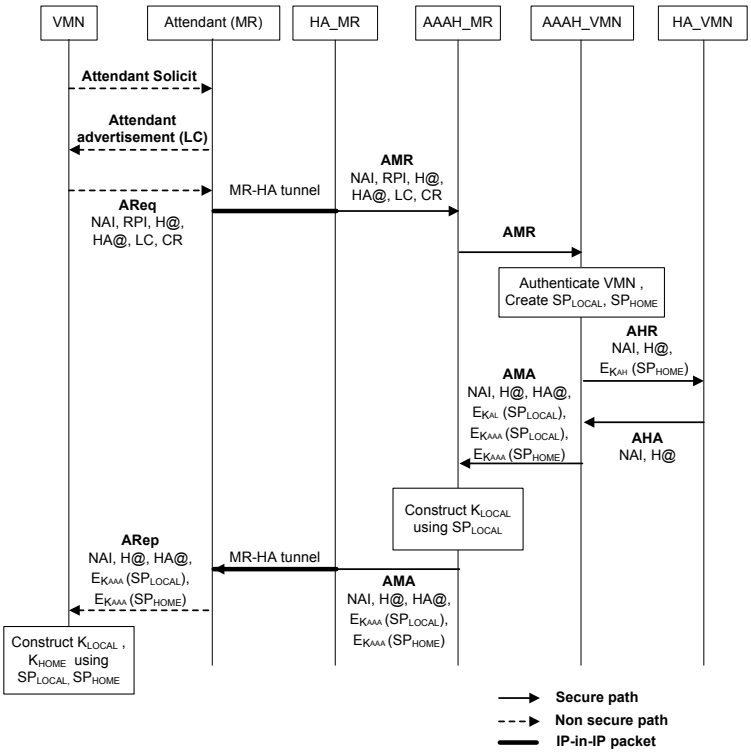


Fig. 7. The AAA porcedure of a VMN

server (i.e $AAAH_{VMN}$) and sends an AReq message to the MR. Then, the MR converts the AReq message into a DIAMETER message, AMR, and sends it to the MR's AAAH server ($AAAH_{MR}$) through a secured bi-directional tunnel. When the $AAAH_{MR}$ receives the AMR message, it sends the AMR message to the $AAAH_{VMN}$ that has a shared SA and requests the AAA procedure for the VMN . Then, the $AAAH_{VMN}$ authenticates the VMN. During this steps, K_{HOME}, K_{LOCAL}, SP_{HOME}, and SP_{LOCAL} are created, similarly to the inter-domain AAA procedure of the MR (see section III-B.1). After completion of AAA procedures, the VMN registers its CoA (configured using the MNP) with its HA.

Only after initial authentication and binding update procedures, VMNs within a MONET do not need to know whether the MONET changes its point of attachment or not. Thus, VMNs do not have to register their locations to their HAs when the MONET hands off. This mobility transparency is the key advantage of the NEMO basic support protocol [8]. However, if the mobility transparency is strictly provided, the AAAL server in the foreign network cannot detect the existence of VMNs. Even if the mobility transparency is beneficial to reduce the binding update traffic, it makes the accounting of VMNs' network usages hard. In our protocol, the AAAL server in the foreign domain accounts the total network usage of the MONET (not individual VMNs) and then this collective accounting information is delivered to the MR's AAAH server. At the same time, the MR's AAAH server maintains the accounting information for the MR as well as individual VMNs[1]. Consequently, the MR's AAAH server can differentiate the accounting information for MRs and VMNs. In addition, we assume that the MR's AAAH server and the VMN's AAAH server have a trust relationship and a shared SA. Therefore, the accounting information collected at the MR's AAAH server is securely transferred to the VMN's AAAH server for suitable billing.

In addition, the mobility transparency causes another problem, i.e. how to authorize VMNs when the MONET moves to a foreign domain with a different billing policy. To solve this problem, an MR sends an `Attendant Advertisement` message with a set R bit when the foreign domain has a different policy and a new AAA procedure is required. Hence, from the `Attendant Advertisement` message, the VMN determines whether it should perform a new AAA procedure or not. In this paper, we assume that each network domain can have different policies, so that the VMN performs a new AAA procedure per inter-domain handoff.

IV. Security Analysis

In this section, we show that the proposed AAA protocol provides mutual authentication. Then, we consider security attacks, e.g. key exposure, replay attack, man in the middle attack.

[1]In the NEMO basic support protocol [8], all packets destined to MNNs are tunneled at the MR's HA, so that the MR's HA can keep track of network usages of individual LFNs and VMNs. We assume that the MR's HA will report this information to the AAAH server.

A. Mutual Authentication

Mutual authentication is a security feature in which a client (i.e. the MR and VMN) must prove its identity to a service (i.e. network), and the service must prove its identity to the client. Therefore, to provide mutual authentication in our protocol, the following requirements should be met.

1) the MR or VMN authenticates the foreign network.
2) the foreign network authenticates the MR or VMN.

Specifically, mutual authentication is achieved as follows.

First, in the case of the inter-domain authentication, mutual authentication is achieved by establishing a session key, K_{LOCAL}. In the other word, the objective of inter-domain authentication protocol is that the MR and the AAAL server believe that they share K_{LOCAL} with each other.

The MR creates CR as

$$CR = E_{K_{AAA}}(LC), \tag{1}$$

where $E_K(\cdot)$ is an encryption function using a key of K.

The AAAH server can verify the MR's identity by comparing with CR sent by the MR and the CR constructed by the AAAH server itself. If two values are equal, the MR is authenticated successfully. Otherwise, the authentication fails. In our protocol, a malicious MR cannot create the correct CR because it does not have K_{AAA}.

After verifying the identity of the MR, the AAAH server transmits $E_{K_{AAA}}(SP_{LOCAL})$ and $E_{K_{AL}}(SP_{LOCAL})$ to the AAAL server through a secure path. When the AAAL server receives, it constructs K_{LOCAL} using $E_{K_{AL}}(SP_{LOCAL})$ and forwards $E_{K_{AAA}}(SP_{LOCAL})$ to the MR. At last, the MR constructs K_{LOCAL} using $E_{K_{AAA}}(SP_{LOCAL})$. After this procedure, the MR and the AAAL server share K_{LOCAL}.

In the case of the intra-domain authentication, the AAAL server in the foreign network verifies the identity of the MR by comparing $E_{K_{LOCAL}}(LC)$ constructed by the AAAL server with CR_L sent by the MR.

On the other hand, to authenticate the foreign network, the MR uses an MC and CR_M. The AAAL server in the foreign network sends CR_M that is created by

$$CR_M = E_{K_{LOCAL}}(MC). \tag{2}$$

Then, the MR can authenticate the AAAL server in the foreign network by verifying that $E_{K_{LOCAL}}(MC)$ is equal to CR_M.

Consequently, a malicious network cannot offer fake services to an MR because it cannot compute CR_M.

B. Key Exposure

K_{AAA} is a pre-shared key between an MR and an AAAH server and K_{LOCAL} and K_{HOME} are created using security parameters (i.e. SP_{LOCAL} and SP_{HOME}). Thus, it is desirable not to leak these keys to the other network entities.

In terms of K_{LOCAL}, the AAAH server encrypts SP_{LOCAL} and sends it to the AAAL server and the MR using K_{AL} and K_{AAA}, respectively. The value encrypted by K_{AL} can be decrypted by the AAAL server, while the other value encrypted by K_{AAA} is decrypted by the MR. Therefore, as K_{AL} and K_{AAA} are not exposed, other entities except the AAAL server and the MR cannot know SP_{LOCAL} and hence cannot construct K_{LOCAL}. As similar to K_{LOCAL}, SP_{HOME} is encrypted using K_{AH} and K_{AAA} and delivered to the HA and MR, respectively. Therefore, K_{HOME} derived from SP_{HOME} is not revealed to other entities except the HA and MR.

C. Replay Attack

Replay attack involves the passive capture of data and its subsequent retransmission to produce an unauthorized effect. A malicious node keeps an `AReq` message and then it can retransmit an old `AReq` message to trick the AAAL server for false authentication. This replay attack can be prevented as follows. LC is created randomly and hence it always changes. Therefore, the malicious node cannot replay the old `AReq` message. When even the same LC can be selected by the attendant by chance, RPI (e.g. time stamp) can prevent the replaying attack.

D. Man in the Middle Attack

A man in the middle attack is that an attacker is able to read, insert, and modify messages between two parties without either party knowing that the link between them has been compromised. In a mobile hotspot scenario, we can imagine an attack that a malicious MR in the middle relays authentication messages and then it intends to use network resource illegally. Figure 8 illustrates the man in the middle attack of a malicious MR in the case of inter-domain authentication. The malicious MR acts as an AR and relays authentication messages between the victim MR and the AR. After the authentication procedures, the malicious MR still can relay all of the traffic between the victim MR and AR. However, the malicious MR cannot use any network resource because it cannot know the K_{LOCAL} and K_{HOME}. Suppose that the object of an authentication protocol is an establishing a fresh session key, the malicious MR cannot compromise the authentication procedure between the MR and the AAAL server.

V. Performance Evaluation

Through the analytical model, we evaluate the AAA cost (C_{AAA}), which is defined the volume of AAA-related messages delivered over the network. Therefore, the unit of C_{AAA} is $bytes * hops$ [13], [14]. Reducing the AAA cost is an important requirement in mobile hotspots where a MONET moves with a high velocity and hence AAA procedures are frequently performed (e.g. train or car). Suppose there are i total handoffs (intra-domain handoffs and inter-domain handoffs) and j inter-domain handoffs for each session. The AAA cost of the MR authentication in the proposed AAA protocol is given by

$$C^{MR}_{AAA}(i,j) = (i-j)\cdot C^{MR}_{intra} + j\cdot C^{MR}_{inter}, \quad i \geq j \tag{3}$$

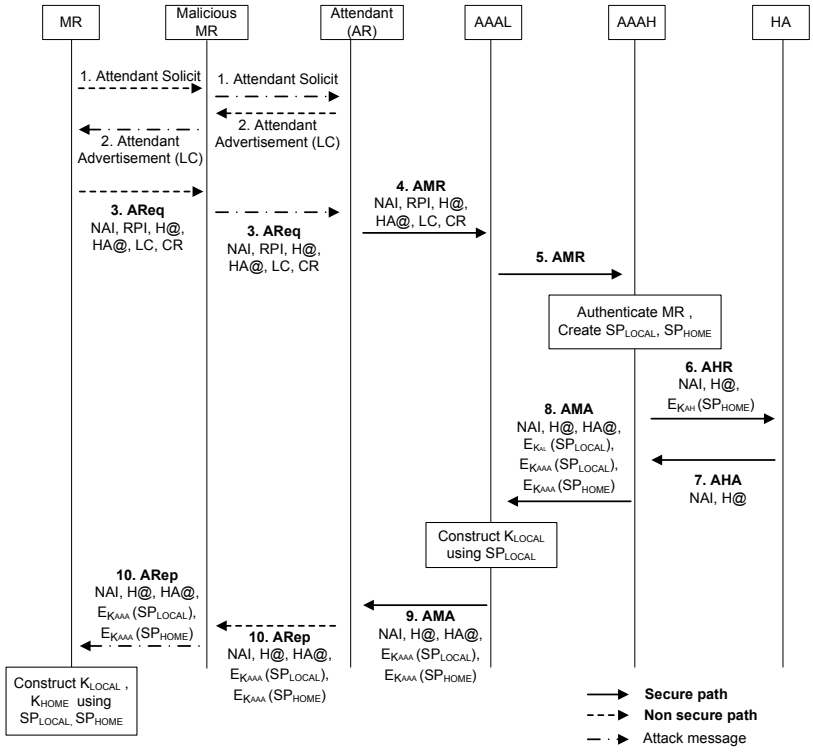


Fig. 8. The scenario of man in the middle attack of the malicious MR

where C_{intra}^{MR} and C_{inter}^{MR} are the costs for intra-domain AAA and inter-domain AAA operations.

The AAA cost of the MR authentication without the localized AAA procedure is given by

$$C_{AAA}^{MR}(i,j) = i \cdot C_{non-local}^{MR}, \tag{4}$$

where $C_{non-local}^{MR}$ is the cost for an AAA operation without the localized AAA procedure.

In this paper, we assume the subnet residence time of the MONET follows a general distribution with a mean of $1/\mu_S$, whose probability density function (PDF) is $f_S(t)$ and its Laplace transform is $f_S^*(t)$. The domain residence time of the MONET follows a general distribution with a mean of $1/\mu_D$, whose PDF is $f_D(t)$ and its Laplace transform is $f_D^*(s)$. When the inter-session arrival time is assumed to be an exponential distribution with a mean of $1/\lambda_I$, the PDFs of i and j are given by [15]

$$\alpha(i) = \begin{cases} 1 - \frac{1}{\rho_S}\left[1 - f_S^*(\lambda_I)\right] & i = 0 \\ \frac{1}{\rho_S}\left[1 - f_S^*(\lambda_I)\right]^2 \left[f_S^*(\lambda_I)\right]^{i-1} & i > 0 \end{cases}$$

$$\beta(j) = \begin{cases} 1 - \frac{1}{\rho_D}\left[1 - f_D^*(\lambda_I)\right] & j = 0 \\ \frac{1}{\rho_D}\left[1 - f_D^*(\lambda_I)\right]^2 \left[f_D^*(\lambda_I)\right]^{j-1} & j > 0 \end{cases}$$

where $\rho_S = \lambda_I/\mu_S$ and $\rho_D = \lambda_I/\mu_D$.

Consequently, the average AAA cost of the MR is given by

$$C_{AAA}^{MR} = \sum_j \sum_i C_{AAA}^{MR}(i,j) \cdot \alpha(i) \cdot \beta(j). \tag{5}$$

In terms of VMN's AAA cost, we consider the AAA cost incurred while the VMN is attached to the MONET. Assume that the VMN's attachment time follows an exponential distribution with a mean of $1/\eta_A$. In addition, let k be the number of inter-domain handoffs during the attachment time. Then, the PDF of k is given by

$$\gamma(k) = \begin{cases} 1 - \frac{1}{\rho_A}\left[1 - f_D^*(\eta_A)\right] & k = 0 \\ \frac{1}{\rho_A}\left[1 - f_D^*(\eta_A)\right]^2 \left[f_D^*(\eta_A)\right]^{k-1} & k > 0 \end{cases}$$

where $\rho_A = \eta_A/\mu_D$.

The AAA cost of the VMN when there are k inter-domain handoffs during the attachment time is given by

$$C_{AAA}^{VMN}(k) = k \cdot C_{AAA}^{VMN}, \tag{6}$$

where C_{AAA}^{VMN} is the cost for each VMN's AAA operation. Then, the average AAA cost of the VMN is expressed as

$$C_{AAA}^{VMN} = \sum_k C_{AAA}^{VMN}(k) \cdot \gamma(k). \tag{7}$$

A. Numerical Results

In this section, we evaluate the effects of mobility and a distance between a foreign network and a home network on the AAA cost (i.e. C_{AAA}^{MR} and C_{AAA}^{VMN}). The numerical results are plotted based on the assumptions introduced in Section V. The parameters and the size of each AAA message are shown in Tables II and III, respectively, based on [10], [17]. The weight for a wireless link is set to 10 [16] and the number of subnets in a domain is set to 49. The distances between an $AAAL$

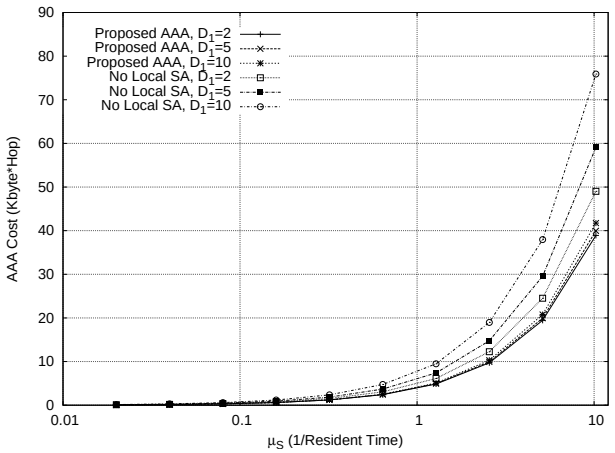


Fig. 9. The AAA cost of an MR

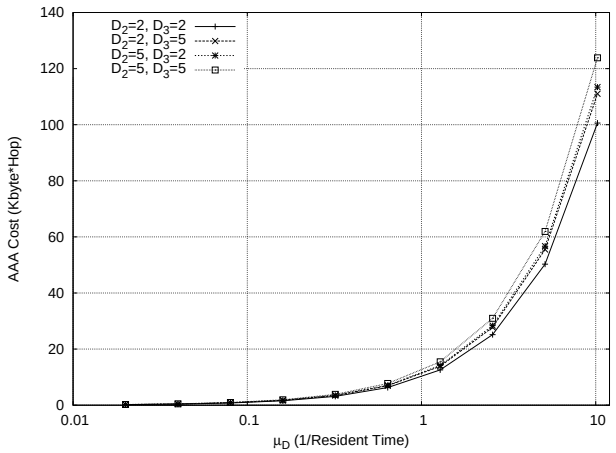


Fig. 10. The AAA cost of a VMN

server and an $AAAH_{MR}$ server, between an MR and its HA, and between an $AAAH_{MR}$ server and an $AAAH_{VMN}$ server are D_1, D_2, and D_3, respectively. λ_I and η_A are normalized to 1.0. By the fluid flow model, μ_D is equal to $\mu_S/\sqrt{N}$ [19].

As shown in Figure 9, C_{AAA}^{MR} increases as μ_S increases (i.e. as the subnet residence time of the MONET decreases). This is because the number of inter- or intra-handoffs is reduced when the mobility (i.e. μ_S) is low. Figure 9 also shows the AAA cost variation for different D_1 (i.e. D_1=2, 5, 10). Since C_{intra}^{MR} and C_{inter}^{MR} are proportional to D_1, C_{AAA}^{MR} increases with increase of D_1. However, in the proposed protocol, the effect of D_1 is not notable. It means that our protocol is effective regardless of the distance between the home network and the foreign network.

On the other hand, if the localized AAA using the K_{LOCAL} is not supported, the MR's AAA cost increases more significantly as μ_S increases. Also, the performance gain becomes more remarkable as μ_S and/or D_1 increase. The effect of D_1 is more clear in the non-localized AAA scheme because an AAA procedure is always performed at the AAAH server in the non-localized AAA scheme. Note that this AAA cost is analyzed for a single MR. As the mobile hotspots services are proliferated, the reduction of the AAA cost (AAA traffic) will be a significant issue.

Figure 10 plots the AAA cost of a VMN, which exhibits a similar trend to Figure 9. Note that the AAA cost when (D_2, D_3) is (5,2) is higher than the AAA cost of (2,5). This is due to IP-in-IP packet tunneling overhead between the MR and its HA. Namely, as D_2, which denotes the distance between the MR and its HA, increases, more tunneling overheads incur and then the AAA cost also increases. As similar to Figure 9, the AAA cost of the VMN is not highly dependent on distance values in our protocol, so that it is concluded that out protocol is less sensitive the distance between the home network and the foreign network.

VI. Conclusion

In this paper, we propose a localized AAA protocol for public mobile hotspots. The proposed AAA protocol is consistent with the NEMO basic support protocol in that the mobility transparency is supported. We analyzed the security concerns in the proposed AAA protocol in terms of mutual authentication, key exposure, replay attack, and man in the middle attack. The central idea behind the proposed AAA protocol is to introduce a shared key between the MR and the AAA server in the foreign network, so that the AAA procedure for the MR in intra-domain handoffs is localized. Performance evaluation results reveal that the localized AAA procedure reduces the AAA traffic significantly in mobile hotspot environment. Furthermore, the localized AAA procedure is less sensitive to the distance between the home network and the foreign network. Although the mobility transparency has the advantage of keeping the VMNs from sending binding update traffic, it causes an accounting problem that the AR cannot know the network usage of VMNs. We suggest that as the MR's HA can keep track of the network usage of individual VMNs, the MR's home network can exchange VMN's accounting information with the AR's network. This way enables a flexible billing mechanism between different domains.

Acknowledgement

This work was supported in part by the Brain Korea 21 project of the Ministry of Education.

References

[1] J. Ott and D. Kutscher, "Drive-thru Internet: IEEE 802.11b for "Automobile" Users," in *Proc. IEEE INFOCOM*, March 2004.
[2] D. Ho and S. Valaee, "Information Raining and Optimal Link-Layer Design for Mobile Hotspots," *IEEE Transactions on Mobile Computing*, to appear, 2005.
[3] OCEAN Project: http://ocean.cse.unsw.edu.au/.
[4] InternetCar Project: http://www.sfc.wide.ad.jp/InternetCAR/.
[5] OverDRIVE Project: http://www.ist-overdrive.org.
[6] IETF Network Mobility (NEMO) Working Group: http://www.ietf.org/html.charters/nemo-charter.html.

TABLE II

PARAMETERS FOR NUMERICAL RESULTS

wireless weight	number of ARs in a domain	λ_I	η_A	D_1	D_2	D_3
10	49	1	1	2, 5, 10	2, 5	2, 5

TABLE III

MESSAGE LENGTHS (BYTES)

Attendant Solicit	Attendant advertisement	AReq	ARep	AMR	AHR	AHA	AMA	AMLR	AMLA
52	84	116	120	172	144	136	166	180	152

[7] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," IETF RFC 3775, June 2003.

[8] R. Wakikawa, A. Petrescu, P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," IETF RFC 3963, January 2005.

[9] T. Ernst and H. Lach, "Network Mobility Support Terminology," Internet draft (work in progress), draft-ietf-nemo-terminology-03.txt, Feburuary 2005.

[10] P. Calhoun, J. Loughney, E. Guttman, G. Zor, J. Arkko,"Diameter Base Protocol," IETF RFC 3588, September 2003.

[11] F. Le, B. Patil, C. Perkins, S. Faccin, "Diameter Mobile IPv6 Application," Internet draft (work in progress), draft-le-aaa-diameter-mobileipv6-04.txt, November 2004.

[12] B. Aboda and M. beadles, "The Network Access Identifier," IETF RFC 2486, January 1999.

[13] S. Lo, G. Lee, W. Chen, and J. Liu, "Architecture for Mobility and QoS Support in All-IP Wireless Networks," *IEEE Journal on Selected Area on Communications (JSAC)*, vol. 22, no. 4, May 2004, pp. 691-705.

[14] A. Stephane and A. Aghvami, "Fast Handover Schemes for Future Wireless IP Networks: A Proposal and Analysis," in *Proc. IEEE 53rd Vehicular Technology Conf. (VTC)*, 2001.

[15] Y. Lin, "Reducing Location Update Cost in a PCS Network," *IEEE/ACM Transactions on Networking*, vol. 5, no. 2, pp. 25-33, February 1997.

[16] J. Xie and I. Akyildiz, "A Distributed Dynamic Regional Location Management Scheme for Mobile IP," *IEEE Transactions on Mobile Computing*, vol. 1, no. 3, July 2002.

[17] T. Narten, E. Nordmark, W. Simpson, "Neighbor Discovery for IP version 6 (IPv6)," IETF RFC 2461, December 1998.

[18] A. Conta, S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)," IETF RFC 2463, December 1998.

[19] X. Zhang, J. G. Castellanos, and A. T. Capbell, "P-MIP: Paging Extensions for Mobile IP," *ACM Mobile Networks and Applications*, vol. 7, no. 2, pp. 127-141, 2002.

Churn Resistant de Bruijn Networks for Wireless on Demand Systems

Manuel Thiele, Kendy Kutzner and Thomas Fuhrmann
System Architecture Group, Universität Karlsruhe (TH)
76128 Karlsruhe, Germany
eMail: {thiele|kutzner|fuhrmann}@ira.uka.de

***Abstract*— Wireless on demand systems typically need authentication, authorization and accounting (AAA) services. In a peer-to-peer (P2P) environment these AAA-services need to be provided in a fully decentralized manner. This excludes many cryptographic approaches since they need and rely on a central trusted instance. One way to accomplish AAA in a P2P manner are de Bruijn-networks, since there data can be routed over multiple non-overlapping paths, thereby hampering malicious nodes from manipulation that data.**

Originally, de Bruijn-networks required a rather fixed network structure which made them unsuitable for wireless networks. In this paper we generalize de Bruijn-networks to an arbitrary number of nodes while keeping all their desired properties. This is achieved by decoupling link degree and character set of the native de Bruijn graph. Furthermore we describe how this makes the resulting network resistant against node churn.

I. Motivation

Today, typical wireless systems are provider based in the sense that there is one central organization that handles authentication, authorization and accounting (AAA) for all their subscribers. This is obvious with systems like GSM or UMTS where the providers manage everything from the AAA backend to the base stations. But this is also true for home WiFi systems. The difference there is that the subscription relation is pushed from the mobile device to the access point (AP). The owners of the APs can control the access to their devices. Conversely, they are legally responsible for the users. Especially, they have to pay for the bandwidth use. As a consequence such systems typically deny roaming between such privately owned APs, thereby significantly reducing the use of such systems.

In this paper we present one building block for wireless network systems that push the subscription relation back into the end-device while maintaining the current pattern of private AP deployment. It aims at peer-to-peer systems consisting of APs and storing accounts for the users. Credits from such an account can be spent for gaining access while roaming.

Such systems harvest the idle resources available at its participants' APs and provide them on demand to their roaming users. Since in peer-to-peer systems there is no (need for a) central organization the participating nodes have to provide all the required resources and functionality to operate the system. As a consequence, there is no extra cost with doing that. I.e. such an approach enables easy and cheap scalability.

Such systems have been proposed in the literature [1][2]. The key challenge in such networks is the security of the AAA-system to avoid free riding [3] and to motivate participants to contribute resources [4]. This is to ensure that a user employs only resources that are equivalent to the resources he or she provides to the system. Note that 'equivalent' does not mean 'equal' (e. g. in terms of bandwidth). So the system can provide a net gain to its users. E.g., if a person that owns (and pays for) an AP with a flat rate provides access to their AP whenever it is idle; this incurs no extra cost for that person. Conversely, roaming users will highly value being granted access to that AP when they urgently need network access.

We extend that work by presenting a novel approach for one building block of these systems. In our system, messages between AAA-modules of participating peers are routed through a specialized overlay network. This overlay network is formed by the participating APs. Since such an overlay is a virtual structure only, the peers are able to create any network topology. But that topology is crucial for the efficiency, robustness and security of the system: Peer-to-peer systems are systems that lack central instances that could establish relations of trust. They can build trust, however, by combining independent sources of trust to increase the trust in a previously unknown peer. Therefore, the topology has to ensure that seemingly independent peers are in fact with high probability actually independent. One way to achieve this, is an overlay topology that routes messages redundantly over independent paths. Besides the gain in security, this increases the robustness of the system.

In this paper, we present a novel variant of the so-called de Bruijn networks. This class of overlay networks provides for several (depending on a parameter of the network) paths which are non-overlapping with a high probability between any two nodes. So far, such networks have been strongly affected by node churn, i. e. nodes joining and leaving the network, potentially ungracefully and at a high rate. Thus de Bruijn networks were thought unsuitable for ad-hoc and peer-to-peer networks where churn is prevalent due to frequent node joins and leaves[5]. Our proposal keeps all beneficial properties of de Bruijn networks while creating churn resistance with low overhead. Its key idea is to extend the de Bruijn graph construction algorithm from graphs with $n = K^L$ nodes to

graphs with an arbitrary number of nodes.

This paper is structured as follows: Section I-A gives an overview of de Bruijn networks. Section II generalizes their construction to an arbitrary number of nodes and describes how routing works in the resulting networks. Section III gives results from a study using a real-world implementation of the so-modified de Bruijn algorithm. Section IV concludes with an outlook to future work.

A. Introduction to de Bruijn Networks

In a de Bruijn-graph each node has a unique identifier consisting of letters of a given alphabet and having a certain length L. The size of the alphabet is called the de Bruijn-basis K and equals the out degree of each node. The length of the identifier is also known as the level of the node or graph.

Each possible combination of the letters of the alphabet with the given length is present in the graph. Because of this, there only exist de Bruijn-graphs with K^L nodes.

In such a graph every node is connected to all those nodes whose identifier can be created by deleting one character on one side and adding a character on the other side. Thus the network functions like a shift register.

On this graph greedy routing can be used to exploit the low diameter. This is done by determining the overlap of the lower end of the current node identifier with the high end of the destination address. Then routing is performed by shifting in the first character of the destination address, after the overlapping area.

Such a network has a constant in- and out-degree of K and a near optimal diameter of $\log_k(N)$ (The optimum [6] is at: $\lceil \log_k(N*(k-1)+1) \rceil - 1$). Additionally there are K different paths from any source node to any target node. An alternative path is selected by routing to the wrong next hop during the first routing step. These paths all have a comparable length and a high chance that they are non-overlapping, i. e. have no more nodes in common than the source and the destination. For example a de Bruijn network with 1000 nodes and K=10 has a path overlap of 3.7% ([6]). The probability of paths overlapping further decreases proportionally to the product of K and the diameter of the graph. This path redundancy is one of the main reasons to use de Bruijn graphs for security relevant systems: An attacker has to subvert all paths to alter a message.

B. Related Work

The idea of de Bruijn networks is older than most recent peer-to-peer networks. It has been used for interconnection of processors in parallel computing, for example [7]. With regard to the usability in peer-to-peer networks, [6] compared many different network graphs. The de Bruijn graph was one of the most favorable, because of its low diameter and constant degree. The resulting network design was called ODRI (for optimal diameter routing infrastructure). At at the same time de Bruijn graphs have been proposed for peer-to-peer networks [8], [5], [9]. Koorde [9] is an embedding of the de Bruijn graph into a Chord [10] network, it therefore inherits many properties from Chord. To gain resistance against node failures, either a number of backup links are introduced or the alphabet size K is increased up to $O(\log n)$. Since the number of possible de Bruijn nodes is larger than the number of nodes present in the network, each Koorde node is responsible for a large number of de Bruijn identifiers. In [8] the de Bruijn topology is used to establish a distributed hash table which strives for probabilistic guarantees for lookups. Since the lookup guarantees are loosened, the routing table maintenance overhead is reduced too. Furthermore, lookup hot spots can be alleviated by means of caching, and key collisions (which may occur e. g. in file sharing networks) can be tolerated at the cost of more lookup steps.

II. Generalization of de Bruijn Networks

As described in section I-A a de Bruijn-graph is only defined for K^L nodes. This would be a tough restriction, because a peer-to-peer-network normally has an arbitrary and permanently changing size. We therefore alter the building rules of the graph slightly, so that it can hold any number of nodes.

The restriction of the graph that all nodes have the same level L (length of the node identifier) is loosened. Without further restrictions, such generalized de Bruijn-networks tend to degenerate under churn, i.e. to accumulate level differences of neighboring nodes. By this, de Bruijn-networks loose their good properties (constant degree, low diameter, high probability of non-overlapping paths). This paper describes a means to circumvent this degeneration.

To prevent this corruption, the maximum level difference between neighboring nodes is set to one. This is a greater relaxation than is theoretically required, because only a level difference of one level in the whole graph is necessary. But the latter rule can only be enforced with global knowledge, which is hard to obtain in the envisioned distributed system. It will be described in the following how it is possible to keep the property of a limited level difference between neighboring nodes in a environment with churn.

In the following sections we will describe the rules for building up, maintaining and routing in a graph with level differences in time and space. The goal of these rules is first of all to avoid unnecessary rebuilds of the graph as changes happen in time. Second, the properties of the graph should be altered as little as possible when nodes of different levels neighbor each other.

A. Linkage of Neighboring Nodes

To achieve these goals, the ID of a node is extended to the right if the level increases. The node is split into K new nodes, one for every possible extension (see figure 2 compare figure 1).

This leads to the rule that the character with the highest significance is always deleted from one node to the next. Additionally on the right as many characters as needed are added. This can be zero, if the level decreases, one, if the level stays the same or two, if the level increases. Because

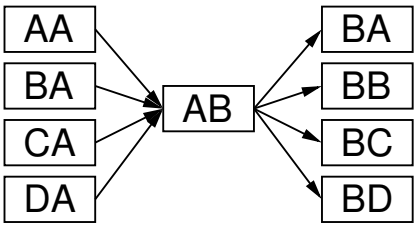


Fig. 1. Part of a Level 2 de Bruijn-Graph with $K = 4$

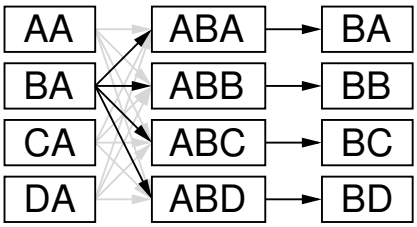


Fig. 2. The ID is extended to the right

of this, the changes of the graph are only local and as many connections as possible are reused.

With these rules the in-degree of a node always stays the same, but the out-degree changes between one and K^2 (reduction of the changes see II-A.1). If the out-degree is of importance (for example for security reasons, there should be always a maximum number of paths to every possible destination) it can be maintained by holding the connections to the former neighbors as irregular neighbors. These are only needed for routing as the first step, when one needs multi-path-routing.

1) Decouple Linking Degree and Character Set: In this scenario K strongly influences the behavior of the network. It sets the connection degree (G) and the number of possible characters (Z).

$$Z = K = G$$

The number of possible characters again defines the number of new nodes, which are formed out of one, when the level of a node changes. The number of new nodes once more causes the decrease or increase of out-degree as shown in the formula.

$$\begin{aligned} &\text{all neighbors have lower level: out-degree} = G/Z = 1 \\ &\text{all neighbors have higher level: out-degree} = G \cdot Z = G^2 \end{aligned}$$

Because Z and G are the same, the out-degree vacillates so strongly. But this strong correlation is not necessary. A much weaker correlation is also possible:

$$G = Z^M$$

being M a arbitrary positive integer.

This replaces only one old character through M new characters, the rest stays the same.

But there is no need to add M characters, if the level of a node is increased. Instead, just one character may be added.

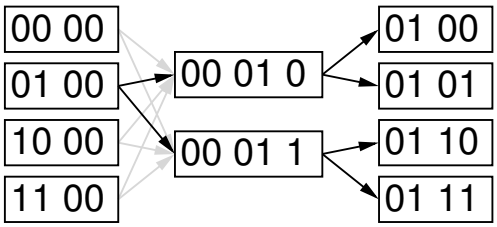


Fig. 3. Decoupling of alphabet and node level.

By this the vacillation is decreased from square to linear. To achieve the best possible gain, Z should be the smallest possible number. The optimum is when $Z = 2$.

Figure 3 shows the idea. The letters of the alphabet are replaced as follows: A=00, B=01, C=10 and D=11.

B. Creating Churn Resistance

The border for level differences of one is hard to obtain in such a network and requires a lot of maintenance effort. This again would cause a low churn resistance.

Secondly for the safety of the data there has to be a certain redundancy of the accounting data through out the network. For these nodes keeping redundant data communication is hard, because they have to use the network, which needs on average $\log_K(n)$ steps.

To overcome these problems, all of the instances which hold the same data for safety reasons only (not those kept apart for security reasons) are brought together in one node (the number of them shall be C). From now on we distinguish between nodes (vertices of the de Bruijn graph) and members of nodes or instances (APs running the implementation). A node can have many members. All these members of one node have the same look of the de Bruijn network and are totally interchangeable. They are connected through a second network for synchronization purposes. This network can vary greatly, it can be a (skip) ring, a (hyper) cube or fully connected. Let the diameter of this network be Innerdim(C,I) depending on the maximum node size and the numbers of connections used.

Such a node has a minimal and a maximal number of members. If the maximum is surpassed the node is split into Z new nodes with a level that is one higher than the old one. If the minimum is reached the split is undone and the Z nodes are joined again into one node.

The number of steps needed for communication between the members of a node is constant because of this. Additionally the numbers of changes in the de Bruijn-network is greatly (depending on the number of node members) reduced. Depending on the threshold values for node splits and joins, the expectation value for a node disappearing at all can be negligible.

By this, the maintenance of the network is much simplified, requires few resources. These are basic requirements for churn resistance.

Because the node size has a (relatively low and constant) upper limit, the effort for the inner node network is almost neg-

ligible (constant). This can justify full connectivity between the members.

As mentioned above, an instance now holds some connections to other de Bruijn nodes and some connections to instances of the same node. Let the upper limit of intra-node connections be I. Therefore an instance still has a constant (maximum) out-degree, but there is a total of $K+I$ connections required. So I and K compete with each other. The total diameter of the network is $\log_K(\frac{n}{C})+\text{Innerdim}(C,I)$ This tern has to be optimized by balancing K and I and maybe C with regard to the churn resistance and the relation of the traffic amount for synchronization purposes and the amount of node to node communication. For example, the target value for I (and thereby node members assuming full connectivity among them) can be set to $O(\log n)$ to create probabilistic survival even if $\frac{1}{2}$ of all instances fail simultaneously.

C. Optimizing Network Properties

In such a network the degree of overlap of paths depends on the level of the node with the lowest level. The diameter again can be estimated by the node with the highest level.

To optimize these and other properties, the level differences should be as small as possible. For this task the fact can be exploited that a node consists of several members. This can be done by not only balancing the node levels but also the node sizes among nodes in the network.

One can use two means:

1) by carefully placing new members
2) by letting members rejoin when differences above a threshold are detected.

In the event of a network with a growing or stable size, the placement of new members through for example random multi point sampling or random walks (see [11]) will be sufficient for balancing the network. Otherwise the number of leaving computers is greater than the one of new computers. Because of this there may be additional balancing needed. For this the neighbors of a node are monitored. If the size of a neighbor (with accounting for the level) is more than one smaller than the own, local members re-execute the placement procedure for new members with special attention to this neighbor. This point of departure is superior to just sending a member to the neighbor because a more global balancing is done and security mechanisms built in this placement procedure can not be circumvented.

The maximum level difference between the node with the highest and the lowest level is

$$\frac{\text{network diameter}}{\text{minimum size of a node}}$$

and kept below this fraction by the eventual node change of individual computers.

III. Implementing and Evaluation

This network design can be easily transformed into a layered architecture shown in figure 4. The lowest layer combines computers or instances to de Bruijn nodes (NMP, Node Management Protocol). If a computer leaves gracelessly, it is detected on this layer.

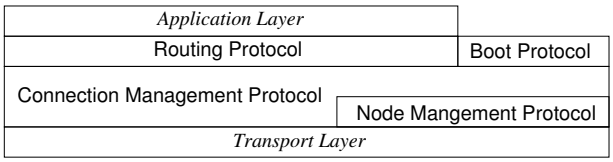


Fig. 4. Protocol stack architecture

TABLE I
Performance of the accounting system (averaged)

maximum number of instances	2186
minimum number of instances	114
lost packets	$4.7 \cdot 10^{-4}$
missing synchronizations	$2.6 \cdot 10^{-4}$
half opened accounts	$3.2 \cdot 10^{-3}$

A second layer assembles nodes to the de Bruijn-network, by handling the neighbors and their knowledge of each other. Additionally, it handles the balancing of the network by detecting differences between neighbors. For all this to work the CMP (Connection Management Protocol) uses the NMP. The routing of messages from an arbitrary node to any other is done by a third layer (RP, Routing Protocol), by using the connections created by the CMP.

The application itself, i. e. the accounting system, also uses the routing protocol to transfer messages between instances of the accounting protocol. It utilizes the offered multi-path-routing for security related data exchange.

At last, the choosing of the integration place for a new computer is done by a fourth protocol (BP, Boot Protocol). This needs to use the CMP, but may also use any of the other protocols.

Such a network, including the accounting system, was implemented in POSIX-C using TCP/IP for the connections[4]. The full source code is available from the authors upon request. Networks with up to two thousend instances of this implementation were tested with respect to functionality and stability under churn. The number of nodes per second, which where disconnected from their neighbors because all of them left before the network could react appropriately, was chosen as metric for the stability of the network. In such extreme cases, the instances execute the boot procedure to re-join the network. The examination of the results confirmed that aggregation of instances to nodes greatly increased the churn resistance of the overall network. Also as expected, the longer members stay in the network, the more stable the network became. The tests showed that with a per instance failure rate of $0.8 \cdot 10^{-3}$ per second a minimum node size of 5 leads to a stable network. For failure rates of $3.2 \cdot 10^{-3}$ per second the node size had to be increased to 8 in a network of 300 instances.

Next we estimated the maximal join and leave speed

our implementation could tolerate without disturbance at the de Bruijn level. We found that with a join rate below two instances per node per second the invariant of maximal level difference between neighboring nodes of one was never violated. Since a rebalance is necessary after instance departures where two nodes are involved, the maximum handled departures rate was one instance per node per second. When these rates are increased, short periods of inconsistencies can occur.

On top of the established de Bruijn routing protocol we implemented an accounting as system designed in [4]. Table I shows the performance tests of the system. Even under the imposed churn (the network shrank by a factor of 20), the system worked as designed. This is due to the stability and churn resistance of the underlaying de Bruijn network. When instances get killed during the reception of a message, this message is lost. In the implemented accounting protocol this leads to missing synchronizations between different account holders and half opened accounts. Both problems can be solved very easily with the repetition of the lost message. In the shown tests these messages were deliberatly not re-sent to show that even without it the system is very stable.

IV. Conclusion and Outlook

In this paper we showed that it is possible to extend the de Bruijn graph from K^L to an arbitrary number of nodes. A network based on such a graph can be an important building block for AAA-systems in the wireless domain, because the resulting network has many desirable properties for security relevant systems. First, the network scales well and has a small diameter, resulting in few hops to, for example, account holders. Second, the structure of the network allows multiple independent paths between any two nodes, which reduces the necessary trust in intermediate nodes. The implementation of the proposal is described together with mechanisms to make the system resistant against node churn.

Further work should explore ways to make the system more resilient against non-arbitrary node failures. Also the ability of the network to recover from net splits or re-joins can be improved.

References

[1] E. C. Efstathiou and G. C. Polyzos, "A peer-to-peer approach to wireless lan roaming", in *WMASH '03: Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, New York, NY, USA, 2003, pp. 10–18, ACM Press.

[2] D. Barkai, "Technologies for sharing and collaborating on the net.", in *Peer-to-Peer Computing*. 2001, pp. 13–28, IEEE Computer Society.

[3] D. Hausheer and B. Stiller, "Peermint: Decentralized and secure accounting for peer-to-peer applications.", in *NETWORKING*, R. Boutaba, K. C. Almeroth, R. Puigjaner, S. X. Shen, and J. P. Black, Eds. 2005, vol. 3462 of *Lecture Notes in Computer Science*, pp. 40–52, Springer.

[4] M. Thiele, "Entwicklung eines sicheren Protokolls für ein Peer-to-Peer-Hotspot–Accounting-System", Feb. 15 2005.

[5] A. Datta, S. Girdzijauskas, and K. Aberer, "On de bruijn routing in distributed hash tables: There and back again.", in *Peer-to-Peer Computing*, 2004, pp. 159–166.

[6] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, "Graph-theoretic analysis of structured peer-to-peer systems: routing distances and fault resilience", in *Proceedings of the SIGCOMM2003 conference*. 2003, pp. 395–406, ACM Press.

[7] M. L. Schlumberger, *De bruijn communications networks.*, PhD thesis, 1974.

[8] A.-T. Gai and L. Viennot, "Broose: A practical distributed hashtable based on the de-bruijn topology", in *Peer-to-Peer Computing*. 2004, pp. 167–164, IEEE Computer Society.

[9] M. F. Kaashoek and D. R. Karger, "Koorde: A simple degree-optimal distributed hash table", in *Proceedings of the 2nd International Workshop on Peer-to-Peer System (IPTPS '03)*, Berkeley, CA, USA, Feb. 2003.

[10] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", in *Proceedings of the SIGCOMM 2001 conference*. 2001, pp. 149–160, ACM Press.

[11] X. Wang, Y. Zhang, X. Li, and D. Loguinov, "On zone-balancing of peer-to-peer networks: analysis of random node join", in *SIGMETRICS 2004/PERFORMANCE 2004: Proceedings of the joint international conference on Measurement and modeling of computer systems*, New York, NY, USA, 2004, pp. 211–222, ACM Press.

Energy-Aware Routing in Sensor Networks: A Large Systems Approach

Longbi Lin, Ness B. Shroff, and R. Srikant

***Abstract*— Sensor network nodes are often limited in battery capacity and processing power. Thus, it is imperative to develop solutions that are both energy and computationally efficient. In this work, we present a simple static multi-path routing approach that is optimal in the large system limit. In a network with energy replenishment, the *largeness* comes into play because the energy claimed by each packet is small compared to the battery capacity. Compared to the other routing algorithms in the literature, this static routing scheme exploits the knowledge on the patterns of traffic and energy replenishment, and does not need to collect instantaneous information on node energy. We also outline possible approaches for a distributed computation of the optimal policy, and propose heuristics to build the set of pre-computed paths. The simulations verify that the static scheme outperforms leading dynamic routing algorithms in the literature, and is close to optimal when the energy claimed by each packet is relatively small compared to the battery capacity.**

***Index Terms*— Energy-Aware Routing, Sensor Network, Large System, Mathematical Programming/Optimization, Simulations**

I. Introduction

Energy-aware routing problem in sensor networks has received significant attention in recent years [10], [11], [16], [18], [20], [21]. Finding a good routing algorithm to prolong the network lifetime is an important problem, since sensor nodes are usually quite limited in battery capacity and processing power. For exactly the same reason, complex routing algorithms do not work well in this scenario, due to excessive overhead. In this work, we are interested in finding a simple and static routing approach. We will also show that under reasonable assumptions, this static routing algorithm suffices: it is optimal in the large system limit.

In our context, the *largeness* comes into play because *the energy claimed by each packet is small compared to the battery capacity.* In this work, we study the routing problem in sensor networks with energy replenishment. Energy sources, e.g., solar cells, can be attached to sensor nodes to prolong the network lifetime [5], [6], [15]. For any individual node, on the one hand, there is energy consumption, which is mainly due to radio communications [1]. There is, on the other hand, incoming energy from the energy source. From this point of view, the battery acts as an energy buffer. We will show the optimality of the static routing algorithm when this buffer size is large, or equivalently, the energy claimed by individual packets is small compared to the battery capacity.

In [12], a dynamic routing algorithm, E-WME, was proposed for sensor networks with energy replenishment. It was shown to be asymptotically optimal when the number of nodes in the network is large. One interesting feature of this algorithm is that it does not need any information on the statistics of the input traffic. The E-WME algorithm is optimal since it achieves a performance ratio (with respect to the best offline algorithm) that is logarithmic in the number of nodes in the network. It is shown in [12] that no algorithm can do better than this algorithm, *if no knowledge about future packet arrivals is present.* However, what if we had some knowledge of the future packet arrivals? For instance, in a sensor network that collects video footage at regular intervals, the data rate may be known a priori. Armed with this kind of information, an algorithm should be able to perform better. In fact, the proposed static routing approach in this paper exploits the available statistical information on the packet arrivals and energy replenishment.

The rest of this paper is organized as follows: in Section II, we formulate the problem of energy-aware routing with energy replenishment, and present our energy queue model. In Section III, we present our algorithm, show its optimality, and discuss the implications. We proceed by discussing some issues related to the implementation of the static approach in Section IV. Numerical results are provided in Section V. Finally, concluding remarks are presented in Section VI.

L. Lin and N.B. Shroff are with Center for Wireless Systems and Applications (CWSA), School of ECE, Purdue University (email: llin@purdue.edu, shroff@ecn.purdue.edu).

R. Srikant is with the Department of Electrical and Computer Engineering and the Coordinated Science Lab, University of Illinois at Urbana-Champaign, (email: rsrikant@uiuc.edu).

II. Problem Formulation

A wireless multi-hop network is described by a directed graph $G(V, E)$, where V is the set of vertices representing the sensor nodes, and E is the set of edges representing the communication links between them. Packets are sent in a multi-hop fashion: a path from source to destination consists of one or multiple edges.

There are I classes of packets. Each class is associated with a different source-destination pair, and possibly different energy requirements for the nodes along the path. Class i packets arrive to the network according to a Poisson process with rate λ_i. For class i, there are $\theta(i)$ pre-computed paths. We use H_{ij}^n to denote the routing matrix: $H_{ij}^n = 1$ if node n is in path j of class i, and $H_{ij}^n = 0$ otherwise. The routing decision on class i packets can be described as

$$\vec{p_i} = (p_{i1}, p_{i2}, \ldots, p_{i\theta(i)}),$$

where p_{ij} denotes the probability of packets of class i being sent along the j^{th} path. We use $\vec{p} = (\vec{p_1}, \vec{p_2}, \ldots, \vec{p_I})$ to denote the total routing decision. In a dynamic routing scheme, p_{ij} can depend on instantaneous information such as residual energy and energy replenishment rates at different nodes, and therefore can be a function of time. In a static routing scheme, pre-computed $\vec{p}$ is used, which is the same as the static splitting probability in a proportional routing scheme.

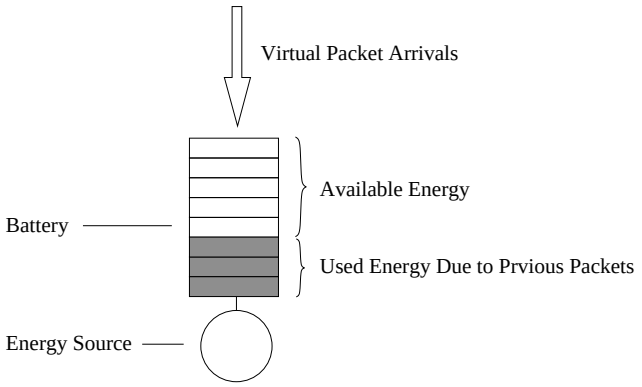


Fig. 1. The energy queue model: one queue case

Each node is modelled as an energy queue, as shown in Figure 1. The battery is a buffer of size K and the nodal energy source serves as the server of the queue. When a real packet is routed through a node (and therefore incurs some amount of energy to replenish), a "virtual packet" arrives to the energy queue associated with this node. Note that our main focus is to model the dynamics of the energy consumption/replenishment processes in the network. Since packet transmissions happen at a much smaller time-scale than the time-scale of energy replenishment, it is assumed that, when a real packet is routed through a path without blocking, a "virtual packet" arrives simultaneously to each of the energy queues along the path. This is illustrated by Figure 2. In other words, there is no notion of packets leaving one energy queue and enter the other. Instead, the interaction of the queues happens through blocking and mean service time, which will be discussed next.

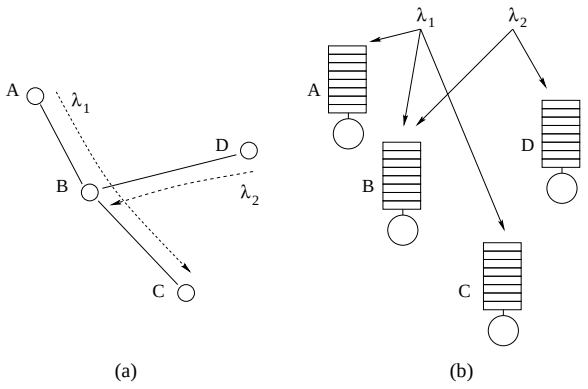


Fig. 2. The energy queue model: multiple queues

If a real packet cannot be sent to the next hop due to energy depletion at one of the nodes, say node n, along the route, this packet is blocked. In a dynamic scheme which utilizes instantaneous system state information, no virtual packet needs to be added to any of the energy queues along the path, since the real packet will be blocked in any case (i.e., there is no gain in admitting the packet). In a static scheme where instantaneous system state information may not be available, the packet may still be relayed down the path until it reaches node n. In this case, only the upstream nodes from node n receive the corresponding "virtual packet" arrivals.

The energy queue is work conserving: as long as there is at least one "virtual packet" in the queue, the energy source will be working on replenishing the used energy due to that packet. The time it takes for node n to replenish the energy consumption due to receiving and/or transmitting a packet of class i sent on path j is *i.i.d* with mean $1/\mu_{ij}^n$. The randomness is mainly due to the stochastic nature of the energy source: for a given energy source, the energy replenishment rate can vary from time to time, even though the overall process is

stationary. [1] Furthermore, different energy sources can have different characteristics in replenishing a certain amount of energy. That is the reason why the service time distribution is assumed to be general. The average energy replenishment rate depends on the following three factors:

- *Node:* Heterogeneous energy distribution is allowed across the network.
- *Class:* Different classes of packets can have different energy requirement for the nodes along the path.
- *Path:* It is assumed that a node can have a complicated power control scheme in which multiple transmission power levels are used to communicate with different neighbors. Therefore, the energy requirement can be different depending on which neighbor it transmits to.

Since the queueing system has finite buffers and the input process is memoryless, under any dynamic policy $\vec{g}$, we assume the system states (the vector of the backlog of all the energy queues) evolve according to a stationary and ergodic stochastic process.

Consider any policy $\vec{g}$. Let $N_{ij}^g(0,t)$ denote the total number of packets admitted to path j of class i during time window $[0,t]$. Define

$$\lambda_{ij}^g = \lim_{t\to\infty} \frac{N_{ij}^g(0,t)}{t} \tag{1}$$

to be the packet arrival rate of path j of class i. This is well defined since the system is stationary and ergodic.

Also let $N_i(0,t)$ denote the total number of class i packet arrivals in $[0,t]$. Clearly, we have

$$\lambda_i = \lim_{t\to\infty} \frac{N_i(0,t)}{t}. \tag{2}$$

Define the average acceptance rate of class i packets on path j to be

$$q_{ij}^g = \frac{\lambda_{ij}^g}{\lambda_i}. \tag{3}$$

Then the total acceptance rate for class i packets is $\sum_{j=1}^{\theta(i)} q_{ij}^g$.

Let $U_i(\cdot)$ be a class-specific utility function of the total acceptance rate of this class. The utility function $U_i(x)$ measures the usefulness of having an acceptance rate of x for class i packets. We make the usual assumption that $U_i(\cdot)$ is strictly concave and non-decreasing for any class.

[1]Note that the energy replenishment process could be non-stationary over very long periods of time, e.g., night and day. However, this can be easily handled by developing different solution for each time of day.

Our goal is to maximize the total weighted utility:

$$\max_g \sum_{i=1}^{I} \lambda_i U_i(\sum_{j=1}^{\theta(i)} q_{ij}^g), \tag{4}$$

subject to energy constraints: a packet can be routed through a path only if all the nodes along the path have sufficient energy (in other words, space in their buffer).

It is worth noting that $q_{ij}^s = p_{ij}(1 - \mathbf{P}_{B,ij})$ for a static scheme, where p_{ij} is the pre-computed splitting probability and $\mathbf{P}_{B,ij}$ is the blocking probability of class i packets on path j.

We will show that a static routing scheme can have performance that is arbitrarily close to that of the best dynamic routing scheme, if the energy claimed by each packet is small compared to the battery capacity. In other words, by carefully designing a static scheme, one can reap most of the benefits of performance without the high cost of implementation.

III. Static Routing with Asymptotic Optimality

In this section, we will show that the performance of a carefully designed static routing scheme is asymptotically optimal in the large system limit. To this end, we first find an upper bound on the performance of the optimal dynamic routing scheme, construct a static routing scheme from the upper bound, and finally show the optimal performance of the static scheme if the energy claimed by each packet is small compared to the battery capacity.

Before stating our main result, we introduce a few notation here. In a system where each energy queue has a buffer of size K, let J_K^* denote the performance (the total weighted utility as defined by Equation (4)) of the optimal dynamic routing, and J_K^s the performance of the static scheme of interest.

Theorem 1: **(Asymptotic Optimality of the Static Routing)**

$\forall \varepsilon > 0, \exists \delta(\varepsilon) > 0$, s.t.

$$\limsup_{K\to\infty} J_K^* < \lim_{K\to\infty} J_K^s + \varepsilon, \tag{5}$$

where the static scheme uses the splitting probability from the solution $\vec{p}$ of the following optimization problem:

$$\max_{\vec{p}} \quad \sum_{i=1}^{I} \lambda_i U_i(\sum_{j=1}^{\theta(i)} p_{ij}), \tag{6}$$

$$\text{subject to} \quad p_{ij} \geq 0, \forall i, j,$$

$$\sum_{j=1}^{\theta(i)} p_{ij} \in [0,1], \forall i,$$

$$\sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \frac{\lambda_i p_{ij} H_{ij}^n}{\mu_{ij}^n} \leq 1 - \delta(\varepsilon), \forall n \in V.$$

Proof of Theorem 1: Please refer to the Appendix for the proof.

Remarks:

1) From Theorem 1, given a set of packet classes, as well as a set of pre-computed paths for each class, a static routing approach can be derived from optimization problem (6) whose total weighted utility approaches the optimal value when the granularity of the battery gets finer and finer. The intuition behind this result is two-fold. First of all, from an energy conservation point of view, the constraints in optimization problem (6) give a fundamental limit on how much utility any dynamic algorithm can achieve, if p_{ij} in (6) is viewed as the average acceptance rate of class i packets on path j under this policy. Furthermore, the static approach using the optimal splitting would be in fact optimal, if there were no blocking, once a packet is assigned to a path. In fact, the probability of such blocking goes to zero, as the per-packet energy consumption becomes smaller and smaller, as compared to the battery size.
2) There are in fact two types of blocking taking place here: (a) the static controller decides from the splitting probability that a packet should be rejected at the source, and (b) a packet is admitted to one of the paths, however, one or more of the nodes along the path in fact does not have enough energy to forward this packet. When we say in the above paragraph that the blocking probably goes to zero, we are referring to the latter case. The reason why it goes to zero is that the ability for the battery to absorb the difference between the incoming and outgoing energy becomes stronger and stronger, as the per-packet energy consumption becomes smaller and smaller. It gives one an illusion that it is the increase in the initial energy that gives rise to the decrease in blocking probability. In fact, in Theorem 1, we do not make any assumption on the initial energy, except that it is certainly upper bounded by the size of the battery. We can assume that all nodes have an empty battery to begin with, and still prove the result of Theorem 1.
3) The convergence of the performance of the static approach to the upper bound is at least as fast as $1/K$, where K is the battery size measured in per-packet energy consumption. This is evident from the second part of the proof in the Appendix.

The static approach is attractive for the following reasons:

- Unlike a dynamic routing algorithm, there is no need to collect information on instantaneous nodal energy. This amounts to a great reduction in routing overhead, which in turn saves more energy in communications. In a practical system where some of the input parameters may be non-stationary (e.g., different average rate of energy replenishment due to seasonal solar radiation), one may need to recalculate the optimal routing probability. Nonetheless, such recalculations can be carried out at a much lower frequency.
- By using the static splitting probability from (6), the static approach adapts to the class-specific utility functions, the traffic load, the topology, and the available in-network energy resources. For instance, different shapes of utility function can lead to different ways of splitting the input traffic. Another example is the energy-aware admission control. If the offered traffic load is quite heavy, with respect to the energy replenishment rate, (6) will produce an optimal solution that is more conservative in admitting the packets. In Section V, we provide some numerical results to further justify the above claims.

IV. Implementing the Static Algorithm

A. Distributed Computation of the Optimal Splitting Probability

To find a way to compute the optimal splitting probability in a distributed fashion, one possibility is to consider the dual of (6). The major challenge here is that the dual function may not be differentiable due to the lack of strict concavity of the primal function. As pointed out in [4], there are at least two ways to handle this problem: one is to use nondifferentiable optimization [3] to solve the dual, and the other is to use the proximal minimization algorithm [13]. We will study the use of both approaches for future work.

B. Obtaining Pre-computed Paths

The proposed static approach is optimal *with respect to* the given set of pre-computed path. Therefore, the quality of the pre-computed paths affects the optimality of the static solution. On the one hand, to maximize the total utility, it is desirable to have a very large set of alternative paths for each class; on the other hand, to lower the overall complexity of the algorithm, we need to limit the number of paths for each class. In fact, there is probably no need to enumerate all the paths between any source-destination pair. Consider the case of a network where the traffic load is relatively uniform. The path that goes through far away nodes probably would not be used even if it was included in the set of pre-computed paths. Therefore, we focus on finding a relatively small set of "good" paths.

Interestingly enough, it is a routing problem by itself to select a relatively small set of "good" paths in an energy-aware fashion. It is then natural to turn to a good dynamic routing heuristic, e.g., E-WME routing [12], to obtain a set of pre-computed paths. The idea is to first cache the paths used by the dynamic routing algorithm for each source-destination pair, and then use the static approach to further optimize on top of the given set of paths.

More specifically, in the sensor networking scenario, one can design a routing setup phase which happens during the deployment of the network. Each node begins by simulating some dynamic routing protocol on the current topology. It is a simulation in the sense that nodes do not pass large data packets, and that they use a virtual battery instead of the real one. When any destination node receives a simulated data packet, it caches the path that the packet has traversed. At the end of this route setup phase, each destination node sends a route summary to the corresponding source node. The static approach can then calculate the routing probability using a distributed solution and take over the routing task.

V. Numerical Results

A. Interaction of Classes: A Simple Example

We now describe the results from our simulations. As described in Figure 3, this network consists of 6 nodes and 6 links. All nodes have the same battery size. It takes unit energy to transmit a packet from one node to another. Table I shows the rate of energy replenishment at different nodes, where $1/\mu_n$ is the average time for node n to replenish the energy due to the transmission of one packet to next hop.

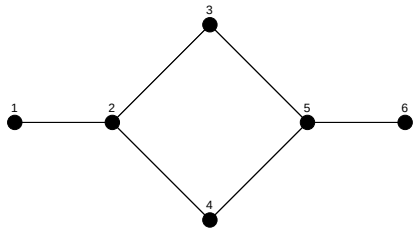


Fig. 3. Example of a small network

TABLE I
RATE OF ENERGY REPLENISHMENT AT DIFFERENT NODES

μ_1	μ_2	μ_3	μ_4	μ_5	μ_6
1.0	1.0	3.0	3.0	1.0	1.0

There are three classes of packets. Each class of packets arrives to the network according to a Poisson process with unit arrival rate. The class-specific concave utility functions is

$$U_i(a) = \frac{\log(t_i a + 1)}{\log(t_i + 1)},$$

where a is the total acceptance rate of class i. The utility function is non-negative, equal to zero if $a = 0$, and equal to one if $a = 1$. The parameter $t_i > 0$ defines the concavity of the utility function: the larger t_i is, the more concave the utility function is.

Table II summarizes the parameters of the three classes and the static solution from optimization problem (6), where δ is chosen to be 0.001. We have the following observations:

- Class 1 and class 3 are symmetric in topology, nevertheless they have different acceptance rates in the static solution. This is because their utility functions are different. The energy available at nodes 2 and 5 is the bottleneck in this scenario, since we only consider energy consumption due to packet transmissions. The way the energy resource, e.g., at node 5, is shared between class 2 and class 3 depends on the shape of their utility function. The utility function of class 3 is more concave. In other words, given any acceptance rate, class 3 has an utility that is greater than or equal to that of class 1 or class 2. Therefore, in the optimal static solution, class 3 has the minimum total acceptance rate, since

TABLE II

PARAMETERS OF THE THREE CLASSES AND THE STATIC SOLUTION

class i	source	destination	t_i	path(s)	static solution
1	1	4	1	$R_{11} = [1, 2, 4]$	$p_{11} = 0.8640$
2	3	4	1	$R_{21} = [3, 2, 4]$ $R_{22} = [3, 5, 4]$	$p_{21} = 0.1350$ $p_{22} = 0.7291$
3	6	4	100	$R_{31} = [6, 5, 4]$	$p_{31} = 0.2699$

its marginal return diminishes faster than the other two.

- The interaction of class 1 and class 2 is through the resource contention at node 2. It turns out they have the same *total* acceptance rate in the static solution, since they have the same utility function.
- All of the acceptance rates are non-zero due to the concavity of the utility functions.

Figure 4 shows that the total utility of the static routing approaches the upper bound as the battery size is increased. Each point of this figure is obtained by running the simulation with different random seeds (the topology remains unchanged) for 100 times and taking the average of the total end-to-end throughput in the steady state. The resulted mean packet data rate is then substituted into the utility function to calculate the total network utility. The upper bound is computed from optimization problem (6), where δ is chosen to be zero.

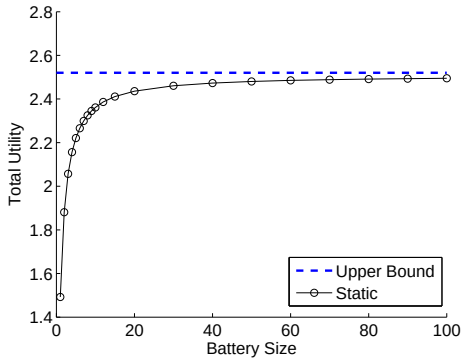


Fig. 4. Total utility as a function of battery size

It is evident from Figure 4 that the total utility of the static solution quickly approaches the upper bound, when the battery size is reasonably large. For instance, if each battery supports transmitting 50 packets without replenishment, the gap between the upper bound and the static approach is less than 1.6%.

B. Throughput Comparison: Static versus Dynamic

For this set of simulations, we randomly deploy 40 nodes on a 10×10 field. Again, all nodes have the same battery size, and it takes unit energy to transmit a packet from one node to another. Furthermore, all nodes have an uniform transmission range (we choose this transmission range to be 3 so that the network is initially connected). There is a link between nodes n and m if and only if (a) the distance between them is less than or equal to the transmission range of a node and (b) node n has enough energy to transmit a packet from n to m directly.

There are 16 classes of packets, each class with a randomly generated source-destination pair. The packets in each class arrive to their source node according to a Poisson process with rate $\lambda_i = 0.6$. For each node, the time it takes to replenish the energy due to transmitting one packet is exponentially distributed with mean $\mu_n = 1$.

We compared the throughput performance of the following three routing approaches:

- E-WME [12] as the dynamic routing approach. The E-WME approach has a built-in admission control component. To decide whether to admit a packet into the network, the E-WME algorithm compares the per-packet revenue to the dynamic E-WME cost metric. Since we want to maximize the throughput performance, we set the per-packet revenue to be a constant.
- Static routing proposed in this paper, with $\delta = 0.001$. The set of pre-computed paths is generated by the E-WME algorithm. For each class of packets, the first 20 paths used by the E-WME algorithm are cached and later passed to the static routing solver to compute the static splitting probability. Since we want to maximize the throughput performance, we set the utility function to be proportional to the total acceptance rate.
- Greedy minimum hop routing. This is a greedy approach in the following sense. On the one hand, this approach tries to take as little energy as possible from the network each time by choosing the path

with minimum hop count. On the other hand, there is no admission control. As long as there is at least one path with enough energy connecting the source to the destination, the packet will be accommodated.

The E-WME algorithm is selected for comparison since it is among the best dynamic energy-aware routing algorithms. In [12], it is shown that it outperforms other energy-aware routing algorithms in the literature. Furthermore, in a competitive ratio sense, the E-WME algorithm is optimal when number of nodes in the network is large.

The greedy minimum hop routing is chosen because it is a natural way to "saturate" the network in order to determine the network throughput capacity, which is limited by the energy replenishment. This should provide a base line approach to which we can compare the more sophisticated static and dynamic algorithms.

We now sketch the static solution by describing the routing decision on class 16, as shown in Figure 5. Class 16 packets travel from node 36 to node 6. As we can see from Figure 5, the static routing solution uses three paths. An incoming packet of class 16 is rejected with probability $P_{\text{reject}} = 0.2062$. An accepted packet is then assigned to one of the three paths with different probability, as indicated in Figure 5. It is interesting to note that the shortest path (shortest in hop count) is not the most preferred path in terms of splitting probability. This is because the routing decision depends on other classes of traffic, in addition to class 16 traffic. The need to load balance here outweighs the importance of using the minimum resources. This is consistent with some observation made in the dynamic routing literature [7], [12], [14] and the online load balancing literature [2].

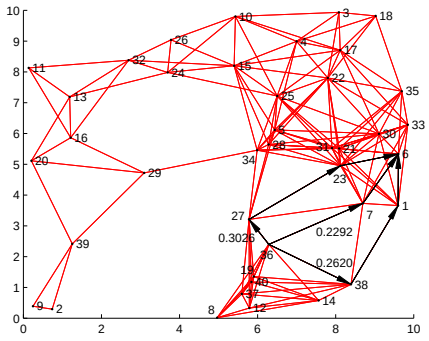


Fig. 5. Multi-path routing solution for class 16

Figure 6 shows the throughput comparison of the three routing approaches as a function of the battery size. The throughput here is the total long-term rate that the network can support, summing over all the 16 classes. Each point of this figure is obtained by running the simulation with different random seeds (the topology remains unchanged) for 10 times and taking the average of the total end-to-end throughput in the steady state. The upper bound is computed from optimization problem (6), where δ is chosen to be zero.

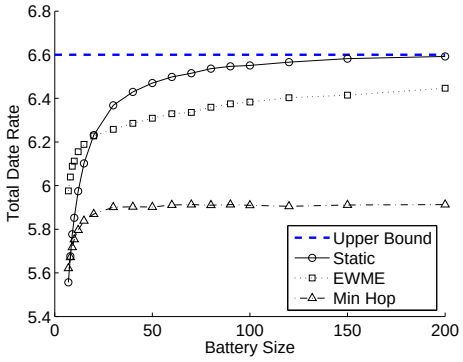


Fig. 6. Throughput comparison of the three routing approaches

It is evident from Figure 6 that the static approach outperforms the E-WME approach and the greedy minimum hop approach, and the throughput performance is close to the upper bound, when the battery size is reasonably large. For instance, if each battery supports transmitting 200 packets without replenishment, the gap between the upper bound and the static approach is less than 0.12%.

There are two major differences between the static algorithm and the dynamic E-WME algorithm:

- They belong to different information regimes [9]. In the static case, we are using statistical information about the input traffic and the rate of energy replenishment at the nodes. In the dynamic case, the E-WME algorithm does not utilize this kind of information explicitly. So one interesting aspect shown by this set of simulation is how much it helps if one knows some information about the traffic, replenishment rate, etc.
- The static approach does not require instantaneous information on node residual energy, which amounts to potentially smaller amount of routing updates.

In setting up the E-WME approach, we use a set of parameters specified in [12]. If we use this routing approach as a heuristic cost metric and further fine-tune the E-WME routing parameters, it is possible to see even better throughput performance than the static

routing. However, this kind of fine-tuning is topology-dependent: different network topology leads to different optimal dynamic routing parameters. Traffic pattern and rate of energy replenishment also have some impact on the choice of the optimal routing parameters. In the static case, this is automatically taken care of, since optimization problem (6) uses the topology/traffic rate/replenishment rate as the optimization constraints.

In general, it may not be important to ask which approach, static or dynamic, is the optimal solution. Rather, it is more about the question of which approach is more suitable for a given scenario. If the application scenario is such that we know some information about traffic/replenishment pattern, but the cost is high to obtain information on instantaneous residual energy, the static approach is preferred. Otherwise, the dynamic approach can be an attractive alternative.

VI. Conclusion

In this work, we address the problem of energy-aware routing in networks with renewable energy sources. Our energy model allows different kinds of energy sources and heterogeneous energy source distribution. The network model allows different classes of packets to have different energy requirement and class-specific utility function, which is defined on the total acceptance rate. We show that a carefully designed static routing algorithm can have performance that is arbitrarily close to that of the best dynamic routing algorithm, if the energy claimed by each packet is small compared to the battery capacity. In other words, by carefully designing a static scheme, one can reap most of the benefits of performance without the high cost of implementation. The results from our simulations confirm the above claim.

The following are possible directions for future work. Section IV-A lists a few options to compute the static solution in a distributed fashion. Instead of using a duality approach, one can also use a primal algorithm with a penalty function [19] to solve the optimization problem approximately. It would be interesting to compare this approach to the proximal minimization solver, in terms of complexity and accuracy.

The performance of the static algorithm approaches the network capacity, which is constrained by the available energy. In addition to taking energy considerations into account, our routing decisions should also take into account different channel conditions, especially in a wireless environment. The goal will be to develop simple and static cross-layer algorithms that favor good channel conditions in order to minimize packet retransmissions, and thus avoid unnecessary wastage of battery resources.

Appendix

Proof of Theorem 1:

(a) Let J_{ub} be the maximum value of optimization problem (6). We first show that J_K^* is upper bounded by $J_{ub} + \varepsilon$.

Let $\tilde{J}_{ub}$ be the maximum value of the follow optimization problem:

$$\begin{aligned} \max_{\vec{p}} \quad & \sum_{i=1}^{I} \lambda_i U_i\Big(\sum_{j=1}^{\theta(i)} p_{ij}\Big), \\ \text{subject to} \quad & p_{ij} \geq 0, \forall i, j, \\ & \sum_{j=1}^{\theta(i)} p_{ij} \in [0,1], \forall i, \\ & \sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \frac{\lambda_i p_{ij} H_{ij}^n}{\mu_{ij}^n} \leq 1, \forall n \in V. \end{aligned} \tag{7}$$

Compared to optimization problem (6), the only difference is that the right hand side of the last inequality is now 1, instead of $(1 - \delta(\varepsilon))$.

Let $N_{ij}^*(0,t)$, λ_{ij}^*, and q_{ij}^* be the corresponding quantities in Equations (1) and (3) for the optimal dynamic scheme. Since

$$\sum_{j=1}^{\theta(i)} N_{ij}^*(0,t) \leq N_i(0,t),$$

from Equations (1), (2), and (3), it is evident that (q_{ij}^*) satisfies

$$q_{ij}^* \geq 0, \forall i, j, \text{ and } \sum_{j=1}^{\theta(i)} q_{ij}^* \in [0,1], \forall i. \tag{8}$$

Furthermore, since each packet in $N_{ij}^*(0,t)$ is eventually served by the server at node n, if $H_{ij}^n = 1$, we apply Little's Law on the server at node n for the class i packets admitted and sent on its j^{th} path:

$$\mathbf{E}\{L_{ij}^n\} = \frac{\lambda_{ij}^* H_{ij}^n}{\mu_{ij}^n},$$

where L_{ij}^n is the in-server queue length of packets from class i, path j. Since the server either processes one

packet when it is busy, or zero when it idles, we have

$$\begin{aligned}
&\mathbf{P}\{\text{Server Busy at node } n\} \\
= \;& \mathbf{E}\{\sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} L_{ij}^n\} \\
= \;& \sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} \mathbf{E}\{L_{ij}^n\} \\
= \;& \sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} \frac{\lambda_{ij}^* H_{ij}^n}{\mu_{ij}^n}. \qquad (9)
\end{aligned}$$

The LHS of the above equation is a probability measure and therefore upper bounded by 1. Since $\lambda_{ij}^* = \lambda_i q_{ij}^*$, we have

$$\sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} \frac{\lambda_i q_{ij}^* H_{ij}^n}{\mu_{ij}^n} \le 1. \qquad (10)$$

From Equations (8) and (10), (q_{ij}^*) satisfies the constraints in optimization problem (7). It follows that

$$J_K^* \le \tilde{J}_{ub}. \qquad (11)$$

We claim that the following relationship also holds:

$$\tilde{J}_{ub} < J_{ub} + \varepsilon. \qquad (12)$$

From Equation (11) and (12), it follows that $J_K^* < J_{ub} + \varepsilon$, which what we want to prove in Part (a).

Now let us show Equation (12) is indeed true. Let $(p_{ij}^0)_{ij}$ be the solution of optimization problem (7). Clearly, $\forall \delta \in (0,1)$, $(1-\delta)(p_{ij}^0)_{ij}$ satisfies the constraint in optimization problem (6). The corresponding function value of (6) is denoted as J_{ub}^0. It follows that

$$J_{ub}^0 \le J_{ub}. \qquad (13)$$

Also,

$$\begin{aligned}
&\tilde{J}_{ub} - J_{ub}^0 \\
= \;& \sum_{i=1}^{I} \lambda_i \left[U_i(\sum_{j=1}^{\theta(i)} p_{ij}^0) - U_i((1-\delta)\sum_{j=1}^{\theta(i)} p_{ij}^0) \right] \\
\le \;& \sum_{i=1}^{I} \lambda_i C\delta \sum_{j=1}^{\theta(i)} p_{ij}^0, \qquad (14)
\end{aligned}$$

where C is a constant. The above inequality is true since each $U_i(\cdot)$ is concave and therefore Lipschitz on the compact constraint set, and there are only finite number of such functions. Furthermore, we can choose $\delta > 0$ small enough such that RHS of (14) is less than any given ε. In other words, we have

$$\tilde{J}_{ub} - J_{ub}^0 < \varepsilon. \qquad (15)$$

From Equations (13) and (15), we are done proving our claim (12).

To sum up, in Part (a), we show that $J_K^* < J_{ub} + \varepsilon$.

(b) Let $(p_{ij})_{ij}$ be the solution to optimization problem (6), then

$$J_{ub} = \sum_{i=1}^{I} \lambda_i U_i(\sum_{j=1}^{\theta(i)} p_{ij}).$$

The performance of the static scheme using $(p_{ij})_{ij}$ as the splitting probability is

$$J_K^s = \sum_{i=1}^{I} \lambda_i U_i(\sum_{j=1}^{\theta(i)} p_{ij}(1 - P_{B,ij})),$$

where $P_{B,ij}$ is the blocking probability of packets from class i, on path j.

If we can show that the blocking probability goes to zero as $K \to \infty$, it follows that $J_K^s \to J_{ub}$. This, combined with Part (a), gives the conclusion in the theorem.

We now show that the blocking probability $P_{B,ij}$ goes to zero as $K \to \infty$.

Let S denote the original system of energy queues with buffer size K, and $\tilde{S}^n$ the following system: node n still has buffer size K, and all other nodes have infinite buffer space. Let $P_{K,ij}^n$ and $\tilde{P}_{K,ij}^n$ denote the probability of the energy queue at node n being full in system S and $\tilde{S}$, respectively.

Since more virtual packets arrive to the queue at node n in system $\tilde{S}$, using a sample path argument, it is clear that $P_{K,ij}^n \le \tilde{P}_{K,ij}^n$.

Let $R_{ij} = (n_1, n_2, \ldots, n_{m(i,j)})$ be path j of class i packets. Let M be the upper bound on the hop count of any path. We define event $\mathcal{B}_{ij}$ to be the event that a packet of class i assigned to path j is blocked, and $\mathcal{F}_n$ to be the event that energy queue at node n full. The blocking probability of class i packets assigned to path j is

$$\begin{aligned}
P_{B,ij} &= \mathbf{Pr}\{\mathcal{B}_{ij}\} \\
&= \mathbf{Pr}\{\bigcup_{n \in R_{ij}} \mathcal{F}_n\} \\
&\le \sum_{n=n_1}^{n_{m(i,j)}} P_{K,ij}^n \\
&\le \sum_{n=n_1}^{n_{m(i,j)}} \tilde{P}_{K,ij}^n \\
&\le M \max_{n \in R_{ij}} \tilde{P}_{K,ij}^n. \qquad (16)
\end{aligned}$$

From the above formula, to show $P_{B,ij} \to 0$, it suffices to show that $\tilde{P}^n_{K,ij} \to 0$. Note that $\tilde{P}^n_{K,ij}$ is the blocking probability of a single $M/G/1/K$ queue with multiple classes of Poisson arrivals. From PASTA [17], $\tilde{P}^n_{K,ij} = \tilde{P}^n_K$, where $\tilde{P}^n_K$ is the probability of queue being full, as observed at an arbitrary time. Therefore, to calculate the blocking probability $\tilde{P}^n_K$, the queue can be viewed as a $M/G/1/K$ queue with a single class arrivals. Consider the corresponding $M/G/1/\infty$ queue, and define the overflow probability

$$\tilde{P}^n_{K,\infty} = \mathbf{Pr}\{Q_n \geq K\},$$

where $Q_n \in \mathbb{Z}^+$ is the workload of energy queue at node n.

A sample path of the workload in $M/G/1/K$ queue can be constructed from a sample path of $M/G/1/\infty$ queue by removing all the time intervals when the workload is above K [8]. It is thus clear that

$$\tilde{P}^n_K \leq \tilde{P}^n_{K,\infty}. \tag{17}$$

Now we have a $M/G/1$ queue with infinite buffer, and we want to show its blocking probability $\tilde{P}^n_{K,\infty}$ goes to zero, as $K \to \infty$. The arrival rate to this $M/G/1$ queue is

$$\tilde{\lambda}_n = \sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \lambda_i p_{ij} H^n_{ij}. \tag{18}$$

A packet from class i on path j has a mean service time of $1/\mu^n_{ij}$, therefore the overall mean service time is

$$\frac{1}{\tilde{\mu}_n} = \frac{\sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \frac{1}{\mu^n_{ij}} \lambda_i p_{ij} H^n_{ij}}{\sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \lambda_i p_{ij} H^n_{ij}}. \tag{19}$$

From (18) and (19), the overall load is

$$\tilde{\rho}_n = \tilde{\lambda}_n \frac{1}{\tilde{\mu}_n} = \sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \frac{1}{\mu^n_{ij}} \lambda_i p_{ij} H^n_{ij}. \tag{20}$$

Recall that $(p_{ij})_{ij}$ is the solution to optimization problem (6). It follows that $\tilde{\rho}_n < 1$. We then invoke Pollaczek-Khinchine formulas [17], and the expected queue length is

$$\mathbf{E}\{Q_n\} = \left(\frac{\tilde{\rho}_n}{1-\tilde{\rho}_n}\right)[1 - \frac{\tilde{\rho}_n}{2}(1 - \mu^2_n \sigma^2_n)], \tag{21}$$

where σ^2_n is the variance of the service time distribution. Since $\tilde{\rho}_n < 1$, $\mathbf{E}\{Q_n\}$ is finite. From Markov Inequality,

$$\tilde{P}^n_{K,\infty} = \mathbf{Pr}\{Q_n \geq K\} \leq \frac{\mathbf{E}\{Q_n\}}{K}. \tag{22}$$

Therefore, $\tilde{P}^n_{K,\infty} \to 0$, as $K \to \infty$. It follows from (17) that $\tilde{P}^n_K \to 0$, as $K \to \infty$.

Q.E.D.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cyirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.

[2] Y. Azar, B. Kalyanasundaram, S. Plotkin, K. R. Pruhs, and O. Waarts. On-line load balancing of temporary tasks. *Journal of Algorithms*, 22(1):93–110, January 1997.

[3] M. L. Balinski and P. Wolfe. *Nondifferentiable Optimization*. North-Holland Publishing Company, Amsterdam, 1975.

[4] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, Englewood Cliffs, NJ, 1989.

[5] DARPA. Darpa energy harvesting program, 2004. Available on `http://www.darpa.mil/dso/trans/energy/overview.html`.

[6] A. Kansal and M. B. Srivastava. An environmental energy harvesting framework for sensor networks. In *Proceedings of the 2003 international symposium on Low power electronics and design*, pages 481–486. ACM Press, 2003.

[7] K. Kar, M. Kodialam, T. V. Lakshman, and L. Tassiulas. Routing for network capacity maximization in energy-constrained ad-hoc networks. *IEEE INFOCOM'03*, 2003.

[8] F. Kelly. Notes on effective bandwidths. *Stochastic Networks: Theory and Applications*, pages 141–168, 1996.

[9] E. Koutsoupias and C. H. Papadimitriou. Beyond competitive analysis. In *IEEE Symposium on Foundations of Computer Science*, pages 394–400, 1994.

[10] L. Li and J. Halpern. Minimum energy mobile wireless networks revisited. In *Proc. IEEE International Conference on Communications (ICC)*, January 2001.

[11] Q. Li, J. A. Aslam, and D. Rus. Online power-aware routing in wireless ad-hoc networks. In *Proc. the Seventh Annual International Conference on Mobile Computing and Networking (ACM Mobicom'01)*, pages 97–107, 2001.

[12] L. Lin, N. B. Shroff, and R. Srikant. Asymptotically optimal power-aware routing for multihop wireless networks with renewable energy sources. In *IEEE INFOCOM'05*, Miami, Florida, March 2005.

[13] X. Lin and N. B. Shroff. An optimization based approach for quality of service routing in high-bandwidth networks. In *IEEE INFOCOM'04*, Hong Kong, March 2004.

[14] S. A. Plotkin. Competitive routing of virtual circuits in ATM networks. *IEEE Journal of Selected Areas in Communications*, 13(6):1128–1136, 1995.

[15] MicroStrain Press Release. Microstrain wins navy contract for self powered wireless sensor networks, December 2003. Available on `http://www.microstrain.com/news_29.htm`.

[16] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333–1344, 1999.

[17] M. Schwartz. *Telecommunication Networks, Protocols, Modeling and Analysis*. Addison-Wesley Publishing Company, November 1988.

[18] S. Singh, M. Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 181–190, 1998.

[19] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, Boston, MA, 2003.

[20] C.-K. Toh. Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks. *IEEE communications Magazine*, 39(6):138–147, June 2001.

[21] J. Wieselthier, G. Nguyen, and A. Ephremides. Energy limited wireless networking with directional antennas: the case of session-based multicasting. In *Proc. IEEE INFOCOM'02*, New York, 2002.

A Synthetic Function for Energy-Delay Mapping in Energy Efficient Routing

Abdelmalik Bachir and Dominique Barthel
France Telecom R&D
Meylan, France
{Abdelmalik.Bachir, Dominique.Barthel}@francetelecom.com

Martin Heusse and Andrzej Duda
LSR-IMAG Laboratory
Grenoble, France
{Martin.Heusse, Andrzej.Duda}@imag.fr

***Abstract*— In traditional approaches to energy efficient routing, a node needs to receive routing messages from all of its neighbors to be able to select the best route. In a previous work, we have proposed a technique that enables the best route selection based on exactly one message reception [1]. Our protocol delays forwarding of routing messages (RREQ) for an interval inversely proportional to the residual energy. Energy-delay mapping techniques make it possible to enhance an existing min-delay routing protocol into an energy-aware routing that maximizes the lifetime of sensor networks. We have proposed some heuristic functions to perform the energy-delay mapping. This paper analyzes their limitations and derives a suitable synthetic function that guarantees that a node selects the best route with very high probability. We also identify comparative elements that help us to perform a thorough a posteriori comparison of the mapping functions in terms of the route selection precision. Simulation results show that our synthetic functions select routes with very high precision while keeping the propagation delay of routing messages reasonable.**

I. Introduction

Sensor networks are composed of wireless nodes that sense various environmental phenomena and maintain communication interconnection via multihop routing. These easily deployable, self-organized, and relatively low-cost networks are expected to be massively deployed in many applications such as habitat monitoring, disaster relief and surveillance [2]–[4]. The success of the applications relies on the network lifetime that depends on the life span of nodes. Hence, energy saving is the crucial factor in designing long-lived sensor networks, mainly because nodes are powered by batteries that may be costly, difficult, or even impossible to replace or recharge.

Designing a universal scheme for optimizing energy savings is challenging due to the variety of sensor network applications. However, for most of applications, measurements presented in the literature [5], [6] and obtained from our expeiments (Table I[1]) show that radio communication is a major source of energy consumption. Therefore, many protocols at different layers have been proposed to address this issue [9]. In the rest of this paper, we focus on energy-efficient routing protocols [10].

At the routing layer, energy-efficient protocols use one strategy or a combination of them to maximize network lifetime: a) min energy metric and b) max-min residual energy metric. In min energy routing, nodes select the route that consumes the least amount of energy. Usually, nodes adjust their transmission power and construct a minimum energy topology to reduce the overall energy consumption of the network [11], [12]. The resulting topology guarantees that each node communicates with other nodes using the route that consumes the least amount of energy possible overall. In max-min residual energy routing, nodes estimate their residual energy and cooperate to prevent the most vulnerable ones from being overused avoiding in this way premature energy exhaustion [13]. Such protocols choose routes bypassing vulnerable nodes, which ensures load balancing and avoids early network fragmentation.

TABLE I
Current Consumption Measurements for FreeScale MC 13192 SARD

Radio Idle (not ready to receive)	0.5 mA
Radio Tx (Transmit)	39 mA (at +4 dbm)
Radio Rx (Receive)	39 mA
MCU (Active)	10 mA
MCU (Partially Active)	8 mA
LED	4 mA
Accelerometer sensors	3 mA

Many research results (see also Section V) conclude that an energy efficient routing protocol that maximizes the life span of a sensor network should combine both min energy and max-min residual energy metrics, because these two approaches are complementary. Indeed, at the beginning of the network life time, the network is dense and nodes have high residual energy: the use of a pure max-min metric may be counter effective—by trying to protect nodes with low residual energy, the max-min metric always selects routes for which the most vulnerable node has the highest residual energy; such a route may actually dissipate more energy than others. So, the min energy metric, which selects the route with the least energy consumption, is a better choice when nodes have enough energy, i.e. their residual energies are larger than a predefined threshold. The max-min residual energy metric should be used to protect nodes with low residual energy, i.e. less than a predefined threshold.

Although such hybrid protocols contribute to better network

[1]We carried out these measurements on the MC 13192 SARD sensor node. The measurements closely match the values announced in datasheets [7], [8].

lifetimes, they still have some drawbacks. In another work [1], we have identified the problem of superfluous routing messages that a node may receive while making the best routing decision. Indeed, in traditional routing protocols with the metrics such as min energy or max-min residual energy, a node needs to receive routing messages from all of its neighbors to be able to select the best route, because the messages contain values required for route selection. We argue that the reception and comparison of all the messages are not needed, since the node eventually selects only one route. To address this issue, we have proposed an approach that enables the best route selection based on exactly one message reception [1]. Our protocol delays forwarding of routing messages (RREQ) for an interval inversely proportional to the residual energy. In this way, the routing message on the best route arrives the first so that the node may ignore the superfluous routing messages that arrive afterwards. Nevertheless, the proposed energy-delay mapping does not guarantee that the selected route is always the best, because the intentional forwarding delay was based on heuristic functions [1].

In this paper, we address these limitations and propose a synthetic function instead of heuristic ones to make sure that a node selects the best route with very high probability. We also identify comparison elements that help us afterwards to perform a thorough a posteriori comparison of the mapping functions in terms of route selection precision.

II. Background

A. Diffusion Routing

Energy-delay mapping techniques enhance any min-delay routing including gradient routing used in *Directed Diffusion* [14]. Gradient routing is destination-initiated in the sense that data collectors (also called sinks) interrogate data publishers (also called sources) asking for specific data. This phase, similar to a route request in on-demand routing protocols, is called interest propagation. It establishes localized data-forwarding pointers (called gradients) from sources to sinks. The sources then stream the requested data back to the sinks according to the directions indicated by the gradients. Although there are different implementations of gradient routing, one phase pull directed diffusion is the best fit when few sinks collect the data published by many sources [15]. Since such situations are fairly frequent in sensor network applications, we consider without loss of generality the one phase pull directed diffusion[2] and enhance it with our solution based on delaying routing messages (RREQ) for an interval inversely proportional to the residual energy.

Our motivations for using diffusion are the following:

- Computational complexity is reduced to a minimum. Each node only needs to broadcast one interest message during the interest propagation phase and it only needs to receive one interest message to setup its routing table (it can ignore the subsequent interest messages related to that same interest). The latter property is particularly interesting, because we have designed a MAC protocol able to identify redundant frames before their complete reception [16]. In this way, a node may turn the radio off to avoid receiving superfluous interest messages, which saves energy.
- There is no overhead due to the exchange of extra information like hello or route metrics messages, which saves more energy and reduces the complexity of the routing protocol in terms of computation and memory occupation. Remind that sensor nodes have usually very limited capacities (for example, nodes used in our experiments have a 8-bit micro controller running at 16 MHz maximum speed and 4KB RAM).
- Routing tables only require one entry per active interest consisting of a pointer toward the next node downstream.
- It enables in-network processing to aggregate data based on attributes used in diffusion, which furthermore saves energy by reducing the size and the number of transmitted/received messages.

[2]which we will simply call diffusion.

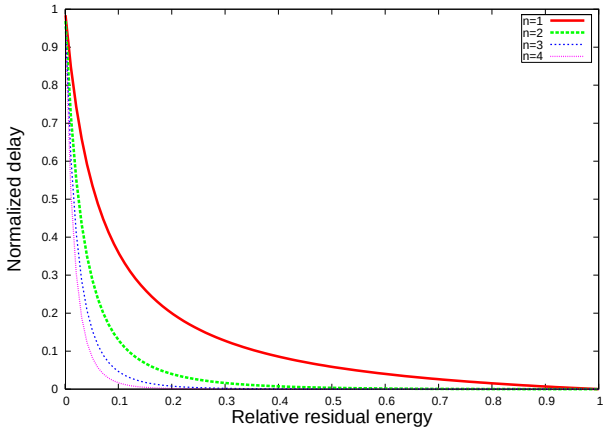


Fig. 1. Heuristic Mapping Functions

B. Heuristic Mapping Functions

Nodes using energy-delay mapping compute a forwarding delay based on their residual energy and defer forwarding of interest messages for this period of time. We have defined energy-delay mapping functions having the property that high residual-energy nodes forward messages without delay, in which case diffusion is equivalent to min energy routing. Nodes with lower energy delay forwarding for a time interval, which results in max-min residual energy routing.

To find a mapping function f with suitable properties, we have explored a family of decreasing convex functions of the form $(1/x)^{\eta}$, where η is a positive parameter. We have shifted and shrunk them so that they map $[0,1] \rightarrow [0,1]$: the residual energy in $[0,1]$ into the normalized delay in $[0,1]$. Fig. 1 presents the resulting set of functions labeled f_{η} with η taking integer values from 1 to 4.

The form of this set of functions can be controlled through two parameters. The first parameter, called sensitivity threshold, separates the min energy metric, when the flat part of the

function is used, from the max-min residual energy metric, when the curvy part of the function is used. For example, the sensitivity threshold of function f_3 is around 0.5, which means that a node using this mapping function does not apply intentional delay when its residual energy is larger than 0.5. Therefore, if we have routes with nodes having residual energies larger than 0.5, the selected route will be the one with the min-delay, which very likely corresponds to the shortest path consuming the minimum energy[3].

The second parameter is the convexity of the function that determines the ability of the mapping function to perform max-min routing. The purpose of the convexity is to have the intentional delay applied by the node with the minimum residual energy on a route being dominant. In this way, the route with the max-min residual energy will be selected, because the routing message of this route will have the smallest delay. The convexity parameter determines the precision of the approximation in Eq. 30 (see Appendix): the more convex the mapping function, the better the approximation. For example, function f_4 has stronger convexity than other functions in the considered set so that it performs better max-min routing.

These heuristic functions have some drawbacks: a) they are likely to be sub-optimal and b) there is a correlation between the convexity of the function and the sensitivity threshold. That is, if we need more precise max-min routing, we will have a smaller sensitivity threshold (e.g. 0.2 for f_4).

To overcome these drawbacks, we propose in the next section a synthetic mapping function that allows exact min to max-min delay mapping according to an uncorrelated predefined threshold. This mapping function is to be used in the situation in which residual energies of nodes are expressed as step functions and not continuous ones.

III. Synthetic Mapping Function

A. System Model

We use the following definitions and assumptions:

- Each node is able to measure its relative residual energy ζ, $(0 \leq \zeta \leq 1)$.
- We call γ the battery protection threshold, $(0 < \gamma < 1)$.
- A node is *vulnerable*, if its residual energy is less than battery protection threshold γ.
- A node is *critical* for a route (to which it belongs), if it has the least amount of residual energy among all the nodes forming that route.
- The *residual energy of a route* is equal to the residual energy of the critical node for that route.
- A *route is vulnerable*, if its residual energy is less than γ.

We assume that there is an *ideal* routing protocol that maximizes the lifetime of a sensor network. According to literature (see Section V), the ideal protocol combines both min energy and max-min residual energy metrics. We assume [4] that the ideal protocol relies on the battery protection threshold concept [17], that is, the ideal protocol uses the min energy metric to select routes as long as there is no any vulnerable route to save energy per packet transmission. Otherwise, when all the routes become vulnerable, the ideal protocol uses the max-min residual energy metric to protect the most vulnerable nodes.

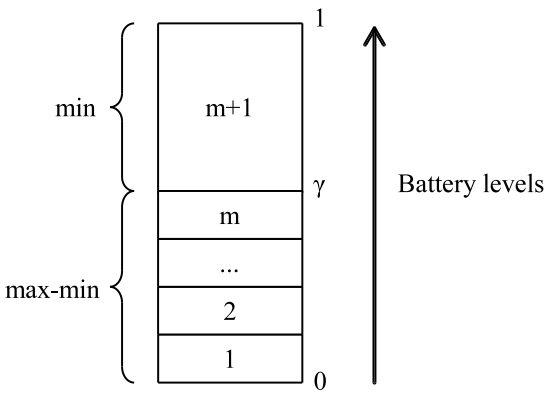


Fig. 2. Energy Levels

In actual implementations of routing protocols, the energy-delay mapping function would likely be discrete and tabulated. Indeed, a node may read its battery voltage or internal resistance and perform table lookup to get the corresponding level of its residual energy. Therefore, we can assume that residual energies of nodes are discrete. We aggregate all the energy levels greater than γ into one energy level as shown in Fig. 2. We call m the number of energy levels that are less than γ. We assign to each node an energy level l depending on its residual energy. We can say that a node with residual energy ζ has energy level l if

$$(l-1)\frac{\gamma}{m} < \zeta \leq l\frac{\gamma}{m}. \tag{1}$$

If ζ is larger than γ, the node has energy level of $m+1$. Explicitly,

$$l = \begin{cases} \left\lceil \frac{m\zeta}{\gamma} \right\rceil & \text{if } \zeta \leq \gamma \\ m+1 & \text{otherwise.} \end{cases} \tag{2}$$

Let g be a synthetic function that maps residual energy into intentional forwarding delay d: $d = g(\zeta)$. As we use discrete energy levels instead of continuous residual energy, function g depends on m. Therefore, intentional forwarding delay $d^{(l)}$ that corresponds to energy level l is the following:

$$d^{(l)} = g_m(l). \tag{3}$$

[3]This is true when nodes use the same transmission power and wireless links have the same error rate.

[4]Note that the question of the ideal routing protocol is still open, since the definition of network lifetime itself is still open. In this paper, we consider the time to partition as the definition of the network lifetime.

TABLE II
NOTATION

p_γ	probability that a node is not vulnerable
$\|\mathcal{R}\|$	number of disjoint routes between the source and the sink
$\|R_k\|$	length of route R_k
n	number of intermediate nodes on the longest route between the source and the destination
$P_{min}(k)$	probability that route R_k is not vulnerable
$P_{maxmin}(k)$	probability that route R_k is vulnerable
P_{maxmin}	probability that an ideal protocol selects a vulnerable route

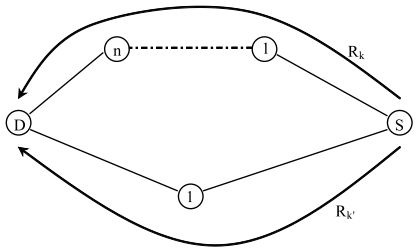


Fig. 3. The Worst Case

B. Deriving Synthetic Mapping Function

Assume we have source node S and destination node D. We call $\mathcal{R}$ the set of all possible routes between the source node and the destination node.

Let us consider route R_k, $(R_k \in \mathcal{R})$. We call $|R_k|$ the number of intermediate nodes on route R_k (source node and destination node are not included). We use the following notation to represent R_k, $R_k = N_{k1} - \cdots - N_{ki} - \cdots - N_{k|R_k|}$, where N_{ki} represents an intermediate node on route R_k.

We propose to derive synthetic function g that meets our goals even in the worst case. It is obvious that g needs to be decreasing to have $g(l) < g(l')$ for all $l > l'$. Besides, g also needs to be convex to mitigate the effect of increasing delay cumulated along longer routes. Fig. 3 shows the worst case example that can be expressed with two routes R_k and $R_{k'}$. Route R_k has the maximum route of length $|R_k| = n$ and residual energy level l, whereas route $R_{k'}$ has the minimum route of length $|R_{k'}| = 1$ and residual energy level $l-1$. In this case, $D^{(R_k)}$, the interest propagation delay on route R_k should be less than $D^{(R_{k'})}$. It is sufficient to have

$$D^{(R_{k'})} = D^{(R_k)} + 1. \tag{4}$$

therefore,

$$\sum_{i=1}^{|R_{k'}|} D_{k'i} = \sum_{i=1}^{|R_k|} D_{ki} + 1, \tag{5}$$

where D_{ki} is the delay incurred by node N_{ki}. Actually, delay D_{ki} is composed of two delays: intentional delay d_{ki} caused by the synthetic mapping function and inherent system delay δ_{ki} that includes computation and transmission delays as summarized in Table IV. For example, in contention-based MACs such as 802.11-inspired MACs, the system delay also includes the maximum jitter, used to alleviate the rate of collisions caused by simultaneous access to the channel:

$$D_{ki} = d_{ki} + \delta_{ki}. \tag{6}$$

In the worst case, nodes on route R_k experience maximum system delays, i.e. $\delta_{ki} = \delta_{max}$ and nodes on route $R_{k'}$ experience minimum system delay $\delta_{k'i} = 0$. Also, all nodes on route R_k have their energy level equal to l, i.e. $d_{ki} = g_m(l)$ for $i = 1, \cdots, |R_k|$ and the node on route $R_{k'}$ has its energy level equal to $l-1$, i.e. $d_{k'i} = g_m(l-1)$ for $i = 1, \cdots, |R_{k'}|$. Therefore, Eq. 5 can be rewritten as:

$$g_m(l-1) = n\left[g_m(l) + \delta_{max}\right] + 1. \tag{7}$$

We set $g_m(m+1)$ to 0. This means that non vulnerable nodes do not apply any intentional delay, so the min-delay routing turns into min-hop routing: we get non vulnerable routes selected according to min-hop routing, which is equivalent to min energy routing as we assume identical links (nodes use the same transmission power and have the same transmission error probability). Therefore, we have

$$g_m(l) = \begin{cases} (n\delta_{max} + 1)\frac{n^{m-l+1}-1}{n-1} & \text{if } \ l \le m \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

IV. Performance Evaluation

A. Methodology

We propose to compare the performance of our protocol based on min-delay routing with energy-delay mapping with the ideal protocol.

Since an energy efficient protocol combines two metrics, min and max-min, we need to use two performance indices for evaluation. For the min metric, we introduce the global gain ratio defined as the global energy consumption ratio between our protocol and the ideal one. In our simulations, this is equivalent to measuring the ratio between the number of hops, because we assume that we do not use transmission power control and links have the same error probability. More specifically, gain $\mathcal{G}$ is defined as:

$$\mathcal{G} = \frac{\sum |R_{our}|}{\sum |R_{ideal}|} \tag{9}$$

where $\sum$ means the sum over all simulation runs.

For the max-min metric, we introduce another performance index: the criticality of a route C. It depends on the residual energy ζ of that route and on the battery protection threshold γ.

$$C = \begin{cases} \zeta/\gamma & \text{if } \zeta < \gamma \\ 1 & \text{if } \zeta \ge \gamma \end{cases} \tag{10}$$

TABLE III
COMPARISON ELEMENTS

	Min (Ideal protocol)	Max-Min (Ideal protocol)
Min (Our protocol)	Case 1	Case 3
Max-Min (Our protocol)	Case 2	Case 4

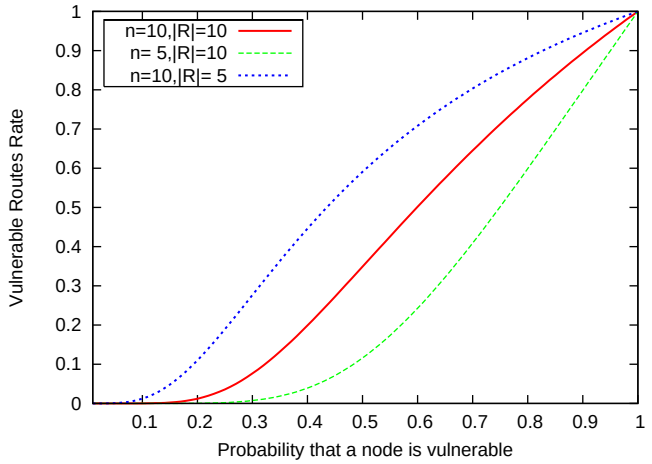


Fig. 4. The Rate of vulnerable routes with respect to the probability that a node is vulnerable. $|\mathcal{R}|$ is the number of disjoint routes between the source and the destination.

We compute the criticality ratio between the criticality of a selected route and the criticality of the ideal route. More specifically, criticality ratio $\mathcal{C}$ is defined as:

$$\mathcal{C} = \frac{\sum C_{our}}{\sum C_{ideal}} \tag{11}$$

Table III summarizes the elements of comparison between our protocol and the ideal one. We distinguish four cases depending on the difference and the resemblance of the metrics used by our protocol and the ideal one.

Case 1: In this case, since both the selected and the ideal routes are not vulnerable (the use of min metric), we only measure the global gain ratio.

Case 2: In this case, the selected route is vulnerable, but the ideal route is not. This case sometimes happens when our protocol fails to find a non vulnerable route, usually when non vulnerable routes are far longer than vulnerable ones. In this case, we measure the average criticality ratio.

Case 3: This case is impossible.

Case 4: In this case, both the selected and the ideal routes are vulnerable (the use of max-min metric). In this case, we measure the average criticality ratio.

B. The Proportion of Vulnerable Routes

We propose to analyze the probability with which a node uses min or max-min metrics to select routes. This probability depends on many parameters shown in Table II.

The ideal protocol picks out a route according to the max-min metric if all the routes are vulnerable. Then,

$$\begin{aligned} P_{maxmin} &= \prod_{k=1}^{|\mathcal{R}|} P_{maxmin}(R_k) \\ &= \prod_{k=1}^{|\mathcal{R}|} (1 - P_{min}(R_k)) \end{aligned} \tag{12}$$

A route is not vulnerable iff all the intermediate nodes on that route are not vulnerable. Therefore,

$$P_{min}(R_k) = \prod_{i=1}^{|R_k|} p_\gamma, \tag{13}$$

where $|R_k|$ is the length of route R_k. So,

$$P_{maxmin} = \prod_{k=1}^{|\mathcal{R}|} \left(1 - \prod_{i=1}^{|R_k|} p_\gamma\right) \tag{14}$$

The mean $E[P_{maxmin}]$ is the following:

$$E[P_{maxmin}] = (E[1 - p_\gamma^L])^{|\mathcal{R}|}, \tag{15}$$

where L is a discrete random variable in $[1, n]$

$$\begin{aligned} E[1 - p_\gamma^L] &= \sum_{i=1}^{n} \left(1 - p_\gamma^i\right) \cdot P\{L = i\} \\ &= \frac{1}{n}\left(n - \sum_{i=1}^{n} p_\gamma^i\right). \end{aligned} \tag{16}$$

Finally,

$$E[P_{maxmin}] = \left[1 - \frac{1}{n}\left(\frac{p_\gamma}{1-p_\gamma}\right) + \frac{p_\gamma^n}{n}\left(\frac{p_\gamma}{1-p_\gamma}\right)\right]^{|\mathcal{R}|} \tag{17}$$

From Eq. 17 and Fig. 4, we conclude that the probability of selecting a route according to max-min (i.e. all the routes are vulnerable) decreases when the number of routes $|\mathcal{R}|$ increases. This means that in dense networks in which there are many alternative routes, finding a route, which is not vulnerable, becomes very likely. We also notice that probability P_{maxmin} increases when the number of intermediate nodes n increases, which is quite expected. Besides, when probability p_γ that a node is not vulnerable increases, probability P_{maxmin} that all the routes are vulnerable decreases, because the number of vulnerable nodes decreases.

C. Worst Case Interest Propagation Delay

Assume that there are n intermediate nodes $N_1, ..., N_n$ between the source and the destination. Each node N_i has residual energy level l_i. On route $R = N_1 - ... - N_n$, node N_i receives the interest at time t_i (we assume the destination sends the interest at time 0):

$$\begin{cases} t_1 & = \delta_1 \\ t_2 & = (g(l_1) + \delta_2) + \delta_1 \\ t_3 & = (g(l_2) + \delta_3) + (g(l_1) + \delta_2) + \delta_1 \\ \vdots & \\ t_{n+1} & = \sum_{i=1}^{n} (g(l_i) + \delta_{i+1}) + \delta_1 \end{cases} \tag{18}$$

TABLE IV

SUMMARY OF DELAYS USED IN SIMULATION

Transmission time	41.6ms (52 bytes at 10kbps)
Computation time	15 to 45ms, uniform
MAC random back-off	0 to 10 * transmission time, uniform

where t_{n+1} is the time when the source receives the interest.

In the worst case, all intermediate nodes N_i, $i = 1, ..., n$ have residual energy levels of 1 (i.e. $l_i = 1$ for all $i = 1, ..., n$) and all system delays $\delta_i = \delta_{max}$ for all $i = 1, ..., n$. Hence, the maximum interest propagation delay in the worst case corresponds to the maximum value of t_{n+1}, which is:

$$\begin{aligned} D_{max} &= n\left(n^{m-1}-1\right)\left(\delta_{max}+\frac{1}{n-1}\right) \\ &= O(n^m \delta_{max}). \end{aligned} \quad (19)$$

D. Simulations

We have run a series of simulations to evaluate the precision of route selection by our protocol based on the proposed energy-delay mapping function compared with the ideal protocol based on the battery-protection threshold. In each simulation run, we have distinguished four cases discussed in Section IV-A. For each case, we have measured the corresponding gain and the criticality ratios. We have also measured the average end-to-end interest propagation delay to evaluate the trade-off between the protocol precision and the delay.

We have carried out 10^4 simulation runs. Each run generates 10 disjoint routes from the source to the sink. To cover a large number of different topologies, we assign a uniformly distributed random number of intermediate nodes to each route. The length of any route does not exceed n intermediate hops. To model the residual energy of nodes, we use the Gaussian distribution $G(\mu, \sigma)$ with mean μ and standard deviation σ. Each node has residual energy distributed according to $G(\mu, \sigma)$; we discard the values of $G(\mu, \sigma)$ outside the interval $[0, 1]$. We have set the battery protection threshold γ to 0.2, because it has been shown that this value results in better performance [18].

As shown in Fig. 4, the rate of vulnerable routes in the network depends on p_γ, the probability that a node is vulnerable, which in turn depends on μ and σ. We have varied μ and σ to compare the precision of our mapping functions, heuristic and synthetic, in different situations. To represent three different situations, we take $\mu = \sigma = 0.5$, $\mu = \sigma = 0.2$, and $\mu = \sigma = 0.1$. We have chosen function f_3, which corresponds to $\eta = 3$ in Fig. 1, as a representative for heuristic functions, because its sensitivity threshold is near the battery protection threshold γ. As a representative for synthetic functions, we take function g_m derived in Eq. 8 with different values for m, $m = 1, \ldots, 5$. For each mapping function, we analyze the precision, evaluated by the gain and criticality ratio parameters, and the average delay to obtain the best precision-delay trade-off. Note that synthetic functions achieve similar precision as the ideal protocol when the best route fits in the min-energy part of routing. Also, Case 2 never happens, because the intentional delay is chosen to avoid this case.

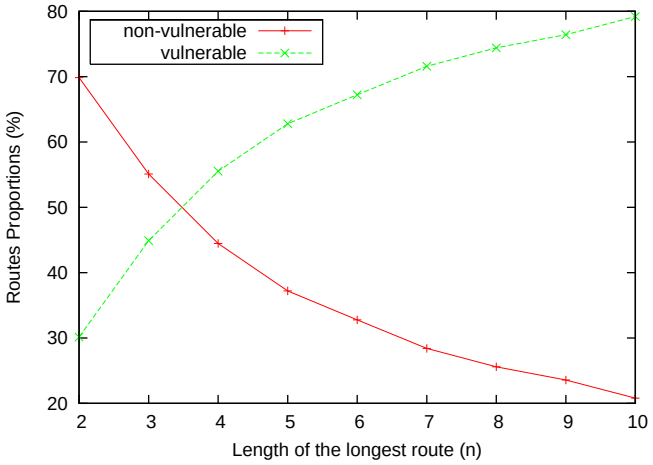


Fig. 5. Routes Proportions

When $\mu = \sigma = 0.5$, there are very few vulnerable routes, around 1%. In this case, we restrict the analysis only to the precision of the min-energy part of heuristic function. Contrary to synthetic functions that select routes with the same length as the ideal protocol, heuristic function f_3 selects routes that are on the average 4% larger than the ideal routes. This is a consequence of the sensitivity threshold of function f_3 being slightly higher that the battery protection threshold. Moreover, function f_3 wrongly uses the max-min metric instead of using the min one for routes with residual energy greater than the battery threshold and less than sensitivity threshold.

When $\mu = \sigma = 0.2$, the proportion of vulnerable routes increases with route lengths from 0.15%, if the maximum route length is 2, to 20%, if the maximum route length is 10. Case 2, in which function f_3 fails to find a non vulnerable route, happens in 4% of the runs. This is due to the convexity of function f_3, that makes the delay on short vulnerable routes not being dominant. Note that we do not have these problems with synthetic functions.

For the precision of vulnerable route selection, function f_3 selects vulnerable routes with a criticality ratio of 95%, which means that residual energy of selected routes is 5% less than the one of the ideal route, whereas synthetic function g_5 selects routes with a criticality ratio of 98%.

When $\mu = \sigma = 0.1$, the proportion of vulnerable routes increases up to 80% for networks with the length of routes up to 10 nodes. Fig. 5 plots the proportion of runs with Case 4 and Case 1, which corresponds to the proportion of vulnerable routes and non-vulnerable routes respectively. We do not plot runs with Case 2, because they are fairly rare, under 2%. In this aging network, function f_3 perfectly selects non-vulnerable routes and selects vulnerable routes with an average criticality ratio of 97%. In this case, the average interest propagation delay is around 2.23 seconds for routes of length up to 10 nodes.

Fig. 6 shows the corresponding criticality ratios and average

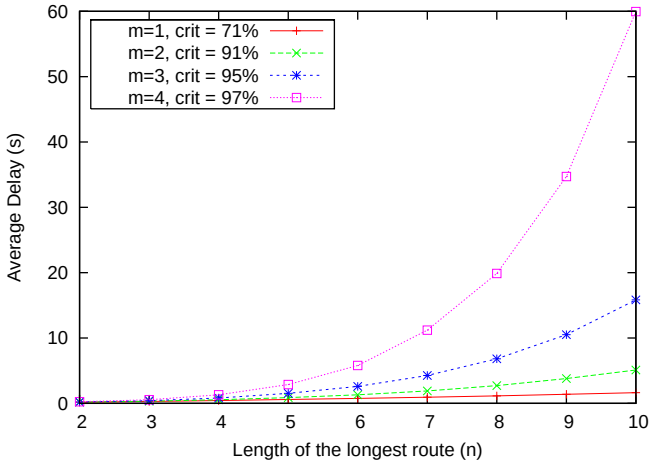


Fig. 6. Average Delay

delays for synthetic function g_m, $m = 1, \ldots, 4$. We show that to obtain to obtain a criticality ratio of 97%, we need four levels of residual energy below the battery protection threshold, *i.e.* $m = 4$. We also show that this criticality ratio requires the average delay of 59.95 seconds. We can conclude that synthetic function g_4 is a good candidate for networks with routes of length up to 10 nodes, as it selects routes with high precision whatever the residual energy distribution of nodes. Indeed, synthetic function g_4 perfectly achieves the min part of the ideal routing and efficiently selects vulnerable routes: only 3% difference between the residual energies of the ideal route and the selected route, which corresponds to the criticality ratio of 97%. We argue that the average delay is not very long because this delay is only used when refreshing routes or finding new ones and it does not affect data delivery latency. We believe that a 1 minute delay to refresh routes is tolerable in network with low dynamicity and steady tasks.

V. Related Work

Toh et al. [17] have proposed CMMBCR (Conditional Max-Min Battery Capacity Routing) for the network lifetime maximization problem. CMMBCR is a combination between MTPR, the min energy metric, and MMBCR, the max-min residual energy metric. In their proposal, they define battery protection margin γ, $(0 \leq \gamma \leq 100)$ and differentiate two kinds of routes: A and Q. Q is the set of all possible routes between a source and a destination nodes. A, a subset of Q, is the set of the routes having residual energy greater than γ, i.e. all the nodes on each route in A have residual energies larger than γ. The protocol is the following: when there is no route in A with residual energy below γ (i.e. all the possible routes contain vulnerable nodes), the protocol selects a route in Q according to the max-min residual energy routing (MMBCR) to protect the most vulnerable nodes. Otherwise, when there is at least one route in A, the algorithm selects a route in A according to the min energy routing (MTPR) to save energy. Note that γ is the parameter that controls the trade-off between MMBCR and MTPR.

Misra et al. [19] take the link transmission cost between nodes into account and propose MRPC (Maximum Residual Packet Capacity) to improve the previous protocol. They model the link transmission cost according to the link error rate and the physical distance between nodes. They introduce a node-link metric C_{ij}, for each link $i \rightarrow j$, that depends on the residual energy B_i of node i, and on the transmission power ζ_{ij} needed to send a packet from i to j. Explicitly, $C_{ij} = B_i/Eij$. The node-link metric determines the lifetime of the link $i \rightarrow j$. The lifetime Life_R of route R depends on the lifetime of the most vulnerable link on this route, $\text{Life}_R = min\{C_{ij}\}$, where $i \rightarrow j$ is a link on route R. The protocol is then straightforward: given a set of routes between a source and a destination node, choose the route with the largest lifetime. Note that basic MRPC is a pure max-min residual energy routing, which could have undesirable behavior by always tending to protect the most vulnerable link. To cope with this issue, Misra et al., propose CMRPC (Conditional MRPC) that uses life protection threshold γ by analogy to the battery protection threshold [17]. That is, CMRPC first tries to select the route with the minimum energy consumption among the routes whose lifetimes are larger than γ. Otherwise, if there is no route satisfying this condition CMRPC switches to MRPC. Simulation results show that CMRPC improves the performance of MPCR, in terms of lifetime maximization, only if the control parameter γ is well determined.

Li et al. [20] address the network lifetime maximization problem with max-min zP_{min}, an on-line message routing protocol. It first computes P_{min}, the minimum energy needed to transmit a packet from a source node to a destination node across all possible routes. It then uses max-min residual energy metric to pick a route, thereby balancing the load among different nodes, unless the cost is higher than zP_{min}, $(z \geq 1)$, in which case, it falls back to the min metric thus avoiding excessive energy consumption. The authors propose a centralized algorithm based on the gradient descent technique to determine the optimal value of z. Further on the same authors describe a distributed version of the algorithm [21], but it requires establishing synchronized mini slots at the MAC layer.

Shah et al. [22] consider the drawbacks of pure minimum energy routing for the survivability of the network. They propose a probabilistic route selection scheme to relieve workload of minimum energy routes. Their protocol is the following: given a set of routes between a source and a destination node, assign to each route the probability of being selected so that the minimum energy route has the highest probability. Then, forward packets on routes according to their probabilities. Note that routes with too much energy consumption, by analogy to the max min zP_{min} algorithm [20], are assigned zero probability and will never be selected. However, this protocol requires to explicitly transmit link cost information and to receive packets from all routes in order to compute the corresponding selection probabilities.

The above papers [17], [19], [21], [22] emphasize the idea of combining the minimum energy and max-min residual energy metrics to optimize the lifetime of sensor networks. However, the distributed nature of these protocols requires explicit transmission of the energy information which is counter productive with respect to energy optimization. Taking this overhead into account and inspired by other papers [23], [24], Guo [25] proposes a lightweight broadcast scheme for network lifetime maximization. His protocol encourages nodes with high residual energy to retransmit a broadcast message and works as follows. When a node receives a broadcast message, it delays the retransmission of this message to see if there is another node with higher residual energy. This delay is inversely proportional to the residual energy of the node. Guo's algorithm reduces the number of nodes forwarding a broadcast message without the overhead of explicitly exchanging the residual energy information, but it may miss some nodes in a sparse network. Besides, it does not implement the minimum energy nor the max-min residual energy routing.

VI. Conclusion

We have presented a synthetic mapping function that enables an existing min-delay routing protocol to be enhanced into an energy-aware routing that maximizes the lifetime of sensor networks. The resulting routing protocol combines the advantages of being min-delay and energy-aware. The energy-aware scheme prevents vulnerable nodes from being overused, which avoids early network partition and the min-delay scheme makes it possible for a node to select routes based on one routing message reception, which avoids consuming extra energy for receiving superfluous routing messages.

References

[1] A. Bachir and D. Barthel. Localized Max-Min Remaining Energy Routing for WSN using Delay Control. *Proceedings of the IEEE ICC*, Korea 2005.

[2] R. Szewczyk et al. Habitat Monitoring with Sensor Networks. *Communications of the ACM*, 47(6):34–40, 2004.

[3] E. Cayirci, T. Cuplu. SENDROM: Sensor Networks for Disaster Relief Operations Management. *MedHocNet*, June 2004.

[4] T. He et al. Energy-Efficient Surveillance Systems Using Wireless Sensor Networks. *Proceeding of the ACM MobiSys*, 2004.

[5] Ember Corporation. EM250 Single-Chip ZigBee/802.15.4 Solution, Data Sheet . 2005.

[6] J. Polastre, R. Szewczyk, D. Culler. Telos: Enabling Ultra-Low Power Wireless Research. *Proceedings of IPSN/SPOTS*, pages 302–11, Los Angeles, CA, April 2005.

[7] Freescale Semiconductor. MC13192/MC13193 2.4 GHz Low Power Transceiver for the IEEE 802.15.4 Standard. *Rev. 2.8*, 04/2005.

[8] Freescale Semiconductor. HSC08 Microcontrollers, MC9S08GB60/D Data Sheet. *Rev. 2.3*, 12/2004.

[9] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. Wiley edition, 2005.

[10] J. N. Al-Karaki, A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, Dec 2004.

[11] Volkan Rodoplu and Teresa H.Y. Meng. Minimum energy mobile wireless networks. *IEEE JSAC*, 8(17):1333–44, August 1999.

[12] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. *Proceedings of the IEEE Infocom*, pages 1388–97, Anchorage, AL, April 2001.

[13] J. Wu, F. Dai, M. Gao, I. Stojmenovic. On calculating power aware connected dominating sets for efficient routing in ad hoc wireless networks. *IEEE/KICS Journal of Communications and Networks*, 1(4):59–70, March 2002.

[14] C. Intagonwiwat et al. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. on Networking*, 11(1):2–16, February 2003.

[15] J. Heidemann, F. Silva, and D. Estrin. Matching Data Dissemination Algorithms to Application Requirements. *Proceedings of the ACM Sensys*, pages 218–29, Los Angeles, CA, November 2003.

[16] A. Bachir, D. Barthel, M. Heusse, and A. Duda. Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks. *submitted for publication*.

[17] Toh, C.-K., Cobb, H. and Scott, D.A. Performance evaluation of battery-life-aware routing schemes for wireless ad hoc networks. *Proceedings of the IEEE ICC*, pages 2824–9, Helsinki, Finland, June 2001.

[18] S. Banerjee, A. Misra. Energy Efficient Reliable Communication for Multi-hop Wireless Networks. *Accepted for publication to Journal of Wireless Networks (WINET)*.

[19] A. Misra and S. Banerjee. MRPC: Maximizing network lifetime for reliable routing in wireless environments. *Proceedings of the IEEE WCNC*, Orlando, FL, March 2002.

[20] Qun Li and Javed A. Aslam and Daniela Rus. Online power-aware routing in wireless Ad-hoc networks. *Proceedings of the ACM Mobicom*, pages 97–107, Rome, Italy, July 2001.

[21] Q. Li and J. Aslam and D. Rus. Distributed Energy-Conserving Routing Protocols for Sensor Networks. *Proceedings of the IEEE HICSS*, Hawaii, January 2003.

[22] R. C. Shah and J. M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. *Proceeding of the IEEE WCNC*, 2002.

[23] S.Y. Ni et al. The Broadcast Storm Problem in Mobile Ad Hoc Network. *Proceedings of the IEEE/ACM Mobicom*, 1999.

[24] T. J. Kwon and M. Gerla. Efficient flooding with Passive Clustering (PC) in ad hoc networks. *ACM SIGCOMM Computer Communication Review*, 32(1):44–56, January 2002.

[25] Xiaoxing Guo. Broadcasting for network lifetime maximization in wireless sensor setworks. *Proceedings of the IEEE SECON*, Santa Clara, CA, October 2004.

Appendix

A. Problem definition

We call ζ_{ik} the relative residual energy of node N_{ik}. Values ζ_{ik} are normalized in $[0, 1]$, hence $0 \leq \zeta_{ik} \leq 1$ for all nodes.

Let us call ζ_k^- the node with least amount of residual energy on route R_k. Then,

$$\zeta_k^- = \min_{1 \leq i \leq |R_k|} \{\zeta_{ik}\} \tag{20}$$

The max-min residual energy routing selects the route with the largest ζ_k^-. Then, max-min residual energy routing selects the route R that satisfies:

$$R = \underset{R_k \in \mathcal{R}}{\operatorname{argmax}} \left\{\zeta_k^-\right\} \tag{21}$$

Combining Eq. 20 and Eq. 21, we get

$$R = \underset{R_k \in \mathcal{R}}{\operatorname{argmax}} \left\{ \min_{1 \leq i \leq |R_k|} \{\zeta_{ik}\} \right\} \tag{22}$$

Let us now examine min-delay routing. We call δ_{ik} the delay introduced by each node N_{ik} on route R_k. Route R_k experiences the total delay of δ_k,

$$\delta_k = \sum_{i=1}^{|R_k|} \delta_{ik} \tag{23}$$

Therefore, min-delay routing selects the route with minimum δ_i. The selected route, denoted by R', satisfies:

$$R' = \underset{R_k \in \mathcal{R}}{\operatorname{argmin}} \{d_k\} \tag{24}$$

Combining Eq. 23 and Eq. 24, we get

$$R' = \underset{R_k \in \mathcal{R}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{|R_k|} \delta_{ik} \right\} \tag{25}$$

Our goal is to make the min-delay routing select the route that satisfies the max-min residual energy metric, *i.e.* route R' matches route R. To make this possible, we propose to use function f to map the residual energies of nodes into an intentional delay. Our goal is to solve Eq. 22 by solving Eq. 25 on a suitable set of

$$\delta_{ik} = f(\zeta_{ik}) \tag{26}$$

B. Approximate Solution: Heuristic Functions

By choosing f to be strictly decreasing, we can rewrite Eq. 22 as:

$$R = \underset{R_k \in \mathcal{R}}{\operatorname{argmin}} \left\{ f \left(\min_{1 \le i \le |R_k|} \{\zeta_{ik}\} \right) \right\} \tag{27}$$

Matching Eq. 27 with Eq. 25 and replacing δ_{ik} by its values calculated in Eq. 26, we conclude that function f such that for all i in $1, \cdots, |R_k|$,

$$\sum_{i=1}^{|R_k|} f\left(\zeta_{ik}\right) = f\left(\min_{1 \le i \le |R_k|} \{\zeta_{ik}\} \right) \tag{28}$$

would meet our goal.

An approximate solution is obtained with f being a convex function $[0,1] \rightarrow [0,1]$. Indeed, if f is convex and decreasing, the minimal ζ_k^- along route R_k makes a dominant contribution to the sum to the left of Eq. 28, i.e. we have

$$f(\zeta_k^-) \gg \left(\sum_{i=1}^{|R_k|} f(\zeta_{ik}) - f(\zeta_k^-) \right), \tag{29}$$

and therefore

$$\sum_{i=1}^{|R_k|} f(\zeta_{ik}) \approx f\left(\min_{1 \le i \le |R_k|} \{\zeta_{ik}\} \right). \tag{30}$$

Evaluation of Energy Heuristics to On-Demand Routes Establishment in Wireless Sensor Networks

Reinaldo C.M. Gomes[1], Eduardo J.P Souto[1,2], Judith Kelner[1], Djamel Sadok[1]
[1]Centro de Informática – Universidade Federal de Pernambuco
[2]Universidade Federal do Amazonas
{ rcmg, ejps, jk, jamel }@cin.ufpe.br

Abstract — **The evolution of wireless communications and embedded system had lead to a big acceptance of Wireless Sensor Networks (WSNs): network of few to hundreds of thousands of sensor nodes, often with low processing and energy capacities that are capable of physical measurements and ambient monitoring, like temperature, humidity and light.**

Considering their limited resources it becomes necessary to support mechanisms that ensure rational use of their resources. It is in this context that we examine routing protocols, and show how these can be energy conscious by avoiding nodes with less spare energy.

Was evaluated six energy based metrics with the objective to make capable the study of some metrics that they can be used for the routes selection in wireless sensor networks and then verify which of them adjust better to the new context introduced in the routing problem by these networks.

Index Terms —**Routing protocols, Energy conservation, Sensor networks, Resource management**

I. INTRODUCTION

The continuous miniaturization of the hardware components joined to the evolution of wireless communications technologies had stimulated the use of Wireless Sensor Networks (WSN) in many applications as environments monitoring, objects tracking and military systems.

This networks were very used as environments monitoring and are often composed by a large number of sensor nodes (hundreds to thousands) with low processing power and energy capacities capable of physical measurements such temperature, light, movement and humidity and convert the collected data in a phenomenon description that can be used to analyze the monitored area.

Different of traditional ad hoc networks, sensor networks equipments have little resources and in the majority of the proposed applications are located in remote areas making difficult the access to maintaining these equipments. In this scenario the network lifetime and the accuracy of the collected data is extremely dependent of the nodes available energy what demands the balancing of that limited resources to allow the functioning of the network for a bigger period and answer to the application specific requirements(i.e. throughput or delay).

To optimize the implementations to this requirement energy conservation techniques must be applied in all stack protocol layers by specific control mechanisms. In the routing layer particularly the main challenge is how to establish energy efficient routes between the nodes and guarantee the delivery of collected data by sensor node to the sink node in order to maximize the functioning time of the network.

Over this scenario is possible see that in sensor networks the routing task is very challenger because of their specifics characteristics and the dependency of the proposed activity for the network. These differences motivate the development of new algorithms that appreciate the new requirements and characteristics of each application to influence in the routes selection process.

The process of search and maintain routes is not minor considering the equipments energy restrictions and the frequents and unexpected network topology changes. To minimize the energy consumption usually techniques as data fusion, data aggregation and nodes clustering are applied by recent researches.

This work presents a sensor network routing protocol family called OPER (On-Demand Power-Efficient Routing Protocol) designed for applications that use a reactive monitoring scheme. OPER was based on AODV (Ad hoc On-Demand Distance Vector) [1] [2] mechanisms for establishing on-demand routes, searching and allocating these as well as DSDV (Destination-Sequenced Distance-Vector) [3] for the control of in order to maintain route table entries.

OPER implements new mechanisms to control route selection using nodes energy information to increase the lifetime of the network. The proposed OPER family can be divided into two algorithm classes considering the mechanisms each one uses to control the residual energy and the routing selection. These two classes are OPER-NE (Node Energy-Aware) and OPER-PE (Path Energy-Aware).

In OPER-NE, a mechanism for route requests acceptance based on the residual energy of each node is applied and

evaluated. The algorithm also uses hop count as metric for route selection.

The main purpose of OPER-NE is allow the control of energy resources in the network nodes in order to avoid they accepting new routing requests when their residual energy cannot attend the lifetime of the solicited new route.

OPER-PE uses the same mechanisms of OPER-NE for route discovery and maintaining as well as the mechanism of selective acceptation of route requests. However, OPER-PE route selection is performed according to the evaluation of heuristics based on the energy state of the nodes in the verified routes as: battery cost and average energy consumption.

Simulation results show that the OPER protocol family offers better results than other routing algorithms evaluated in this work in all the analyzed scenarios. An evaluation score (behavior score) is another contribution of this study and offers a normalized value to simplify the set of metrics evaluated and analyzed.

The rest of this paper is structured as follows: the section 2 discusses some related works in wireless sensor networks routing protocols. Section 3 presents the functioning overview (messages, phases and routing table) of the proposed protocol. The section 5 presents the simulation setup and sixth section shows the simulation results. Finally the 7th section shows the conclusions and future works to be developed.

II. Related Work

Sensor networks introduce new challenges that need to be dealt with as a result of their special characteristics. Their new requirements need optimized solutions at all layers of the protocol stack in an attempt to optimize the use of their scarce resources [4] [5].

In particular, the routing problem, has received a great deal of interest from the research community with a great number of proposals being made. The proposed protocols often resort to the use of artifacts such as data aggregation, nodes clustering and location information.

The majority of these routing protocols can be classified in basically four main classes based in [6]: Data centric, hierarchical, location-based and Network Flow and QoS awareness.

Data centric algorithms are based on the use of network queries where the collected data is named to allow the nodes to search and get only the desired information. This technique is used to avoid the transmission of redundant data in the network and hence saves the network unnecessary work and energy. Two of the main algorithms are Direct Diffusion [7] (that each node disseminate the data interested in receive) and SPIN [8] (meta-data information are transmitted between the nodes to identify the nodes to who send the collected data).

Hierarchical algorithms separate the nodes in subregions called clusters in order to segregate the areas of the monitoring environment as LEACH [9], PEGASIS [10] and TEEN [11]. To allow communication between the clusters a leader is selected from each cluster (cluster-heads). Leaders are then responsible for the management (data aggregation, queries dispatch) and transmission of the collected data in the region they control.

Location-based algorithms (i.e. GAF [12] and GEAR [13]) rely on the use of nodes position information to find and forward data towards a destination in a specific network region. Position information is usually obtained from a GPS (Global Positioning System) equipment.

Finally, network flow and QoS awareness algorithms uses network traffic models and apply QoS based mechanisms to support their routing requirements as SAR [14] or SPEED [15].

In energy routes allocation is founded few works and in the majority it's not evaluate the sensor networks context, analyzing scenarios where these kind of mechanisms are applied in Ad-hoc networks, what does not reflect the same conditions because of the great differences between the resources restrictions of the equipments of each type of network.

Two of the main works in this evaluation of energy based routes selection are [16] and [17] where are presented some metrics to choose path between the network nodes but as said previously are not contemplated sensor networks restrictions making an evaluation that does not demonstrate the efficiency of such mechanisms for networks with scarce resources and without processing capacity to the evaluation of very complex metrics.

III. Evaluated Algorithm

The evaluated algorithm (OPER – On-Demand Power-Efficient Routing Protocol) is a propose presented in [18]. Its consider the sensor networks restrictions, a problem that can be pointed and many of the existing routing algorithms do not answer completely because not evaluate nodes energy parameters in routes selection leaving of appreciate important information about network energy status that allow a better adequacy to the applications requirements.

To allow an increase in the network lifetime the addition of mechanisms in routing protocols to verify another parameters set beyond the hop count that accept a more intelligent routes establishment for the nodes residual energy conditions. This task is done by the applied routes selection process that uses energy based heuristics to adopt it self better to network power consumption than the most of sensor networks routing algorithms.

The algorithm was developed to answer to this objective over the implementation of a power-efficient routing protocol that disseminates the collected data considering nodes energy state.

Next will be presented the main information about the algorithm like messages main phases and routing table one time that are responsible for data forwarding through the network and will be used to exchange information and maintain the created routes.

A. Messages

OPER protocol uses four messages to allow the communication between the nodes: hello, route request, route

reply and route error.

1) Hello Message: The hello message is transmitted always that a node enters in the network to help the neighbors discovery process executed during the network startup. These messages are also transmitted periodically to check topology changes.

2) Route Request Message: This message is used in route establishment process. This process is started when a local entry table is not found for the required route. The network nodes disseminates the request in order to discover a given route.

3) Route Replay Message: Is generated when a node (required destination or an intermediary node) knows of a route that reaches a given destination. The route reply message data is used to create a new entry into the local routing table of origin node.

4) Route Error Message: Is used to signal that no route to the requested destination can be allocated. Another nodes in the network may take advantage of this message to remove any routing reference to this destination previously stored in the routing table.

B. Algorithm Phases

Three phases are responsible for data forwarding through the network. These are based on the previously introduced messages to exchange information between the nodes and the management and maintenance of existing routes.

1) Neighbor Discovery: Before sending data to the sink node, a node must start the neighbors discover process to create a neighbors list that is the address of all nodes that it is able to communicate directly. This information is used to forward packets to a destination and to check for network topology changes.

During this process, hello messages are asynchronously exchanged by the network nodes. Periodically these messages are broadcast to verify nodes reachability and then maintaining the neighbor list up to date. On the receiver of the hello message the sender address is added in its neighbors list and the message is removed from the network. If some neighbors still not transmitting after a period its address is removed from the neighbors list.

2) Routes Discovery: Once the neighbor discovery is terminated the node can initiate the routes discovery when it need establish a route to communicate with the sink node. The presented algorithm adopt an on-demand avoiding the large cost of establish a complete routing infrastructure ready for use at any moment although it is not necessary.

Route Discovery starts with the broadcast of a route request message (RREQ) by the originating node. This message therefore reaches all the neighbors. Upon the receipt of a RREQ, a node performs one of the following actions:

- Send a route replay message (RREP) if it has a path leading to the target destination in its routing table.
- Retransmit by broadcast the RREQ to further nodes (its neighbors).
- Discard the message if the node has already received this request with a better value to the metric, or that its energy is below a threshold stipulated by an application.

3) Route Maintenance: Route maintenance takes two steps: the maintenance of local connectivity achieved through periodic update of neighboring nodes list as well as the maintenance of routes established between nodes.

Hello messages are used to maintain a node's list of neighbors updated. The second step consists of checking whether next hop neighbors maintained their connectivity.

This process also supports route failure notification to all the nodes that use a given route until all the sources using this route are informed of the problem.

C. Routing Table

After established the routes between the network nodes their will be store in a routing table (Table 1) to allow future queries for the allocated paths. The routing table store information about the paths that can be used to direct data messages and verify the validity of each table record.

Table 1. Routing Table

Fields	Description
Destination	Destination Address
Destination Sequence Number	Control Validity Sequence
Next Hop	Next node Address
Hop Count	Hop Count to Destination
Lifetime	Route Validity

These route table also will be updated periodically to reflect the time to invalidate the route entry if it isn't in use and to adapt to the occurred changes in the network.

D. Path energy heuristics

Are used heuristics that evaluate metrics related to the energy state of the nodes that make up a given path the verify the functioning of each one and then analyze the benefits bring by the evaluation of energy based paths and the better set of heuristics for sensor networks.

Six energy based heuristics was implemented: MAER (Maximum Average Energy Routing), MBCR (Minimum Battery Cost Routing), MERVR (Major Energy per Route Validity Routing), MMAER (MMAER - Min-Max Average Energy Routing), MMBCR (Min-Max Battery Cost Routing) and MMERVR (Min-Max Energy per Route Validity Routing).

1) MAER: Hop count based route selection does not offer the best approach when it comes to energy saving within a sensor network. It does not consider the residual energy left within a node when choosing a path and lacks adapting its decisions to changes in the energy map of a sensor network.

In order to recover from these limitations, this work implements the heuristic MAER as route selection metric. That takes into account residual energy of all the nodes that make up a path to compute the average residual energy for each node n_i, *Energy(i)*, over a route *R* with *D* hops between origin and a destination, as shown by equation 1.

$$AVG = \frac{\sum_{i=1}^{D-1} Energy(i)}{D-1} \quad (1)$$

This energy average value is obtained for all possible routes between the origin and destination and the one with maximum energy average is selected as shown in equation 2. In other words, the algorithm opts for using the route that has more overall energy.

$$R_i = \max_{i \in Routes} (AVG) \quad (2)$$

So far this solution may present a serious drawback as it is. The fact that the average energy value over a path is the highest does not imply necessarily that all its nodes have a satisfactory level of residual energy. Actually, one may have a mix nodes with very low and nodes with very high levels of energy. In such extreme cases, the average path energy may also consequently be higher than all the others. One way for dealing with this problem, would be to adopt a second constraint where a path with nodes below a given energy threshold will not be selected.

2) MBCR: Minimum Battery Cost Routing. The second heuristic implemented is the MBCR. That is a proposed routing algorithm based in the functions proposed in [16] and [17] that considers residual energy as a metric for selecting its routes. It associates a cost to each route to a given sink.

Since the inclusion of a node into the path is determined by its residual energy level, the lower this one is the bigger its cost is hence turning it less likely to be selected. The adopted cost function is given by equation 3.

$$C_i = \frac{Total\_Energy}{\mathrm{Re}sidual\_Energy} \quad (3)$$

Where *Total_Energy* is the total node battery capacity and *Residual_Energy* represents its current residual energy of each node. The decrease of *Residual_Energy* results in cost increase. The cost of a route from node *j*, Rj, with Dj is established according to equation 4.

$$R_j = \sum_{i=0}^{D_j-1} C_i \quad (4)$$

Hence in order to obtain a route with the highest residual energy or lowest cost, equation 5 is used to select a route among all the possible ones with minimum cost.

$$R_i = \min (R_j) \quad (5)$$

3) MERVR: The third heuristic implemented in this work is the MERVR. That evaluate the energy that each node can offers during the route validity for a received request. To that the residual energy of all the nodes that make up a path is divided by the route validity established (equation 6).

$$Time\_Energy = \frac{Energy}{Route\_Validity} \quad (6)$$

Once verified this value the obtained result of all nodes are added to calculate the validity energy of the route *D* nodes, as shown by equation 7.

$$Route\_Energy = \sum_{i=1}^{D-1} Time\_Energy \quad (7)$$

This value is obtained for all the path between the origin and destination and the one with maximum score must be selected using the equation 8.

$$R_i = \max_{i \in Routes} (AVG) \quad (8)$$

4) Min-Max: The last OPER-PE implemented heuristic was the Min-Max Battery Routing (MMBR), proposal initially as a refinement of the minimum battery cost heuristic (MMBCR [16] - Min-Max Battery Cost Routing). The initial proposal was extended to allow the use of the maximum average energy (MMAER - Min-Max Average Energy Routing) and major energy per route validity (MMERVR), enabling the adequacy evaluation of both heuristics with the establishment of a minimum threshold of nodes energy in the selected routes.

This algorithm chooses the route that have the greater energy average greater or the lesser battery cost, calculated through Equations 2, 5 and 7, respectively. With this the best routes will be selected since that all nodes who compose it possess a residual energy value above one determined threshold.

Therefore, this metric always tries to prevent routes that possess a lower residual energy of all the possible found routes and expects that the energy of each node will be used more correctly than in the other presented heuristics, preventing the overload of the nodes.

IV. Simulation Setup

The TinyOS [19] platform was used to carry out the simulations. This is a sensor networks development environment that can be used to build and test applications for sensors of the MICAMOTES [20] platform. It also can perform experiments using the TOSSIM simulator also available under this platform.

The values used to build the energy model for the simulations correspond to those for Mica2 sensors and were obtained from [21] and was evaluated the six routing selection energy heuristics presented.

A. Evaluated Metrics

Four metrics were established for the purpose of comparing the algorithms presented in this work.

1) Packet Delivery: The first metric that was analyzed in this work is the packet delivery. This is seen as the fraction of the sink node received messages by the total of messages actually sent by the sensor nodes. Packet delivery provides us with an idea of how efficient a routing algorithm has been in its use of network available bandwidth.

2) Lifetime Duration of Failing Nodes: A second important metric is that of life duration of a node. Power limitations as well the difficulty in reloading nodes with new energy remain considerable obstacles. As a result, a routing algorithm should use as little as possible network resources. As far as the experiments conducted in this work, only average nodes lifetimes were considered for those that were switched off.

3) Number of Failed Nodes: The third evaluated metric is the number of nodes that do not make it to the end of the simulations under each routing algorithm. This information is presented in the form of a fraction.

4) Behavior Index: Finally a composite metric (behavior index) combining the previous three metrics is also used. The idea is to give an overall view of the benefits of each of the examined routing algorithms. This last metric allows for a comparison that takes into consideration the average node lifetime, packet delivery rates and the number of nodes that switch off. The idea is to find a balance between these metrics.

Equation 9 shows the used approach to establishing the composite metric. The smaller its value the better the behavior of the algorithm being analyzed.

$$BI = \frac{\left[\left(\frac{Initial\_Energy}{Life\_Duration}\right) * Failed\_Nodes\right]}{Packet\_Delivery} \quad (9)$$

V. Simulation Results

Next, the simulation results are presented showing the performance of each of the routing algorithms using the metrics described earlier. To claim good accuracy for our study, 100 replications were conducted giving our results a 95% confidence level.

A. Packet Delivery

Fig. 1 shows that depending of the based heuristic of each implementation we can find different functioning related to the packet delivery rate.

Implementations based on the routes cost evaluation (MBCR and MMBCR) present intermediate values and practically didn't suffer to the influence of the network increase maintained an almost constant delivery rates throughout the simulated scenario.

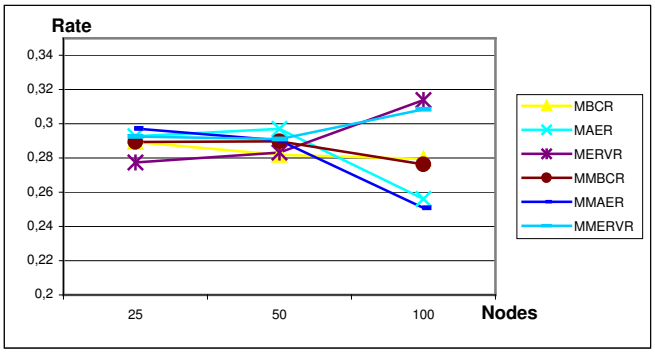


Fig. 1. Packet Delivery Rates.

The MAER based implementations suffers had been the most affected by the network nodes increase. Initially this implementations has presented the better results in the 25 nodes networks but with the increase in the amount of nodes their delivery rate reduce significantly decreasing almost five percentile points (equivalent more than 10% of reduction).

Finally MERVR based implementations presented the better functioning of the heuristics: even don't presenting the best values in small networks these heuristics has adapted better to the network growth and present an increase in the packet delivery rate in the largest networks.

B. Lifetime Duration of Failing Nodes

In Fig. 2 is possible see the obtained results by the algorithms in terms of nodes lifetime duration.

Every algorithms present a tendency of decrease in the nodes lifetime with the increase in the amount of nodes deployed in the network. This decrease depicted in Fig. 2 is a result of their use of message broadcast during route discovery computation process.

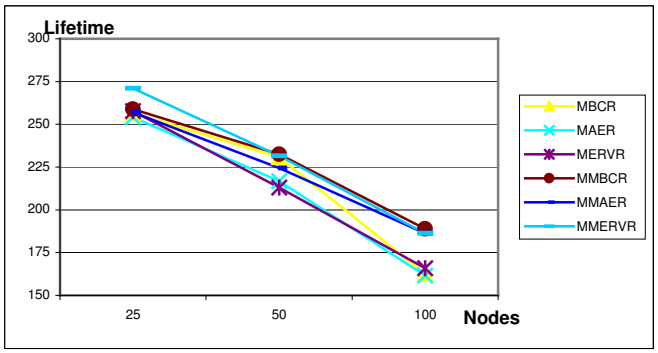


Fig. 2. Average lifetime of failing nodes.

Is possible see that the implementations that uses Min-Max heuristic had obtained higher values for node lifetime than the others implementations even presenting the same decrease trend. The average gain presented by these implementations was bigger than 5% in all the topologies.

C. Number of Failed Nodes

The number of failed nodes in the simulations (Fig 3) present the same trend founded in the nodes lifetime: the implementations that evaluate Min-Max heuristic has obtained better results (with gains greater than 10%) and consequently deactivated less nodes than the implementations of the "pure" heuristics.

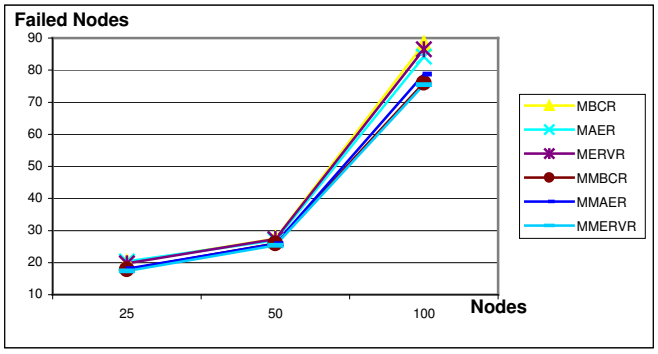


Fig. 3. Number of failed nodes.

Despite it is possible to verify one better adequacy of the heuristics based in the available energy per validity of the routes: implementation MMERVR was the best one between that uses the Min-Max heuristic and the MERVR the best one considering the ones that do not evaluate it.

D. Behavior Index

As expected the values of the BI (Fig. 4) are better to the implementations that evaluate the Min-Max heuristic showing that the addition of more mechanisms to control the energy of the allocated routes enable a better functioning of the algorithms in view to answer the sensor networks requirements.

This implementations obtained an average gain of 15% when compared to the implementations that do not uses the Min-Max heuristic in the routes selection evaluated metric.

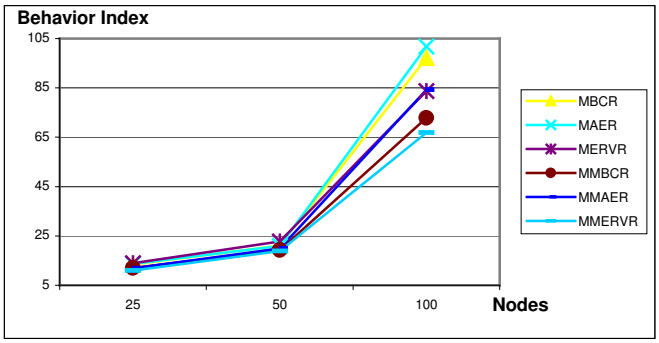


Fig. 4. Behavior Index.

However an unexpected result was the excellent behavior index obtained by the MERVR heuristic that presented similar results to MMAER heuristic on the network increase, allowing therefore a similar network functioning demanding a lesser nodes processing charge.

VI. Conclusion

We can verify that the selected set of information to be used in the route discovery process to evaluate the possible path between the nodes cause a significant difference in the functioning of the network.

This set of information (that base the heuristics to route selection) must be selected considering the nodes restrictions to analyze if the processing charge to be submitted is acceptable. Another point to be considerate is the applications requirements what can be adapted better to the particular characteristics of some heuristic.

As future work we can point to the evaluation of another energy heuristics to routes selection to refine the presented evaluated mechanisms.

Another evaluation to be carried consists of the creation of algorithms for multiple routes allocation. This technique could also be considered in order to improve overall delay, mainly in situations with highly fail of the used routes and to avoid route discovery allowing considerable energy savings in such scenarios.

References

[1] C. Perkins and E. Royer , “Ad hoc On-Demand Distance Vector Routing”, 1999, WMCSA, 234-244.

[2] C. Perkins et al., “Ad hoc On-Demand Distance Vector (AODV) Routing”, July 2003, RFC 3561.

[3] C. Perkins and P. Bhagwat , “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, 1994, ACM SIGCOMM, 234-244.

[4] Akyildiz, I., Su, W., Sankarasubramaniam, Y. e Cayirci, E. (2002) “A Survey on Sensor Networks”, IEEE Communications Magazine, pp. 102-114.

[5] S. Tilak, N. B. Abu-Ghazaleh and W. Heinzelman, “A Taxonomy of Wireless Micro-Sensor Network Models”, Mobile Computing and Communications Review, Vol. 6, No. 2, pp. 28-36, 2002.

[6] K. Akkaya, and M. Younis, “A Survey on Routing Protocols for Wireless Sensor Networks”, 2004.

[7] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", MobiCom'00, 2000.

[8] W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks”, MobiCom’99, 1999.

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks", Hawaii International Conference System Sciences, 2000.

[10] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power Efficient GAthering in Sensor Information Systems", IEEE Aerospace Conference, 2002.

[11] A. Manjeshwar and D. P. Agrawal, “TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks", 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, 2001.

[12] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing", MobiCom’01, 2001.

[13] Y. Yu, D. Estrin, and R. Govindan, “Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks”, UCLA Computer Science Department Technical Report, 2001.

[14] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie, "Protocols for self-organization of a wireless sensor network”, IEEE Personal Communications, Vol. 7, No. 5, pp. 16-27, 2000.

[15] T. He et al., “SPEED: A stateless protocol for real-time communication in sensor networks”, International Conference on Distributed Computing Systems, 2003.

[16] C. K. Toh, “Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks”, IEEE Communication Maggazine, 2001.

[17] M. Maleki, K. Dantu, and M. Pedram, "Power-aware Source Routing Protocol for mobile Ad Hoc Networks", ISLPED’02, 2002.

[18] R.Gomes, J.Kelner, E. Souto, D. Sadok, “Evaluating Energy Mechanisms for Routing in Wireless Sensor Networks”, Mobility 2005, *To Appear*.

[19] TinyOS: a component-based OS for networked sensor regime. http://www.tinyos.net

[20] Mote Hardware Session, Crossbow Technology Inc. http://xbow.com/Support/Support_pdf_files/Motetraining/Hardware.pdf

[21] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen and M. Welsh, “Simulating the Power Consumption of LargeScale Sensor Network Applications”, SenSys2004. 2004.

Blocking Expanding Ring Search Algorithm for Efficient Energy Consumption in Mobile Ad Hoc Networks

Incheon Park, Jinguk Kim, Ida Pu
Department of Computing, Goldsmiths College
University of London, London SE14 6NW, UK
Email: {map01ip, ma201jgk, i.pu}@gold.ac.uk

***Abstract*— Efficient energy consumption is one of the important issues in Ad Hoc Networks. This paper identifies the inefficient elements during the route discovery in well known reactive protocols such as DSR and proposes the new *Blocking-ERS* approach, which demonstrates a substantial energy efficiency at small time expense in comparison to conventional Expanding Ring Search methods.**

I. Introduction

Mobile Ad-Hoc Networks (MANETs) provide an alternative model of communication to conventional wired networks during malfunction or absence of the fixed wired network[1]. They have shown the potential of making significant impacts in situations where the physical infrastructure was either dysfunctional or unavailable, for example, in emergency rescue operations, and in combat zones.

In MANETs, nodes cooperating for delivery of a successful packet form a communication chanel consisting of a *source*, a *destination* and possibly a number of *intermediate nodes* without any fixed base station. Each node communicates directly with the destination or neighbour intermediate nodes within wireless transmission range.

MANETs are hardly realised in practice unless energy efficient communication are in place due to limited battery life of mobile devices. Protocols determine how the nodes communicate and the existing protocols can be modified to become more energy efficient. It is known that the communication between nodes consumes substantial amount of energy in wireless networks [2].

This paper examinates inefficient elements in well known reactive protocols such as DSR [1] and AODV [3] and proposes a new approach for rebroadcasting in Expanding Ring Search. This leads to the *Blocking-ERS* scheme, as we call it, which demonstrates improvement in energy efficiency at the expense of route discovery time in comparison to conventional ERS. We assume that routing protocol can be carried out much faster than the topological changes [4].

The rest of this paper is organised as follows. Section II shortly discuss the existing Expanding Ring Search. Section III describes the general behaviour of TTL sequence-based expanding ring search. Section IV outlines the proposed approach in terms of time delay and energy consumption. Section V discusses the results of this research along with future directions.

II. Expanding Ring Search

Reactive routing protocols in MANETs such as DSR and AODV are often supported by a so-called *Expanding Ring Search* (ERS for short). During the route discovery stage, the RREQ (Route REQest packet) is broadcasted by flooding and propagated from one intermediate node to another to find the route information from the source to the destination node. Figure 1, 2 and 3 show how the broadcasts and propagation form searching 'rings' in such a route discovery process.

In Figure 1, a RREQ is broadcasted by the source and two neighbour intermediate nodes receive the message. Each arrow line represents a send-receive relationship between the broadcasting source and one neighbour node. Each broadcast is issued with a *hop number* which is a serial number indicating the sequence of the nodes alone a route from the source. In Figure 2, for example, a RREQ is broadcasted with a hop number '1' by the source and five intermediate nodes receive the message and are given the same hop number. If none of them has the route information to the destination node, the five nodes rebroadcast the RREQ with an incremental hop number, 2 in Figure 3, for example. In this way, the nodes with the same hop number from the source node form a circle, i.e. the search rings. As the route discovery in progress, the diameter of the searching ring increases.

Utilising the route cache during routing is widely adopted in MANETs to achieve time and energy saving routing. A node in MANETs broadcasts RREQs and the intermediate nodes within the broadcast range cooperate the route discovery process by checking their own route caches for requested route

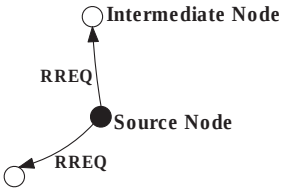


Fig. 1. Propagation of RREQs

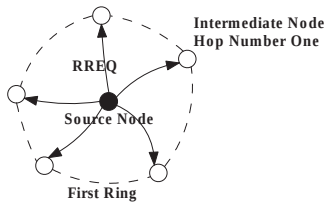


Fig. 2. Formation of a ring

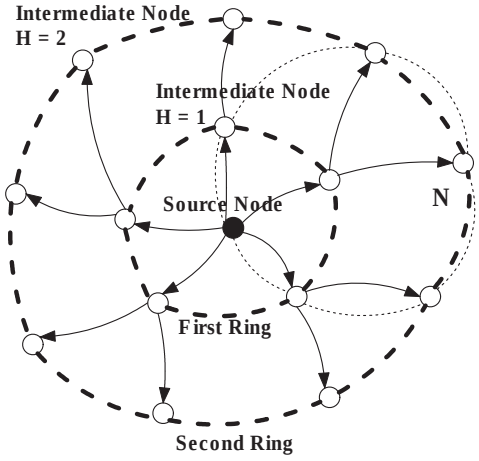


Fig. 3. Expanding search ring

information and maintaining an updated list of known routes. An intermediate node who has the requested route information towards the destination is defined as a *route node* in this paper.

Efficient route discovery relies largely upon how quickly a route node can be found. When a RREQ is received, an intermediate node searches for the requested route information in its route cache. If there is route information to the destination node in its route cache, the route node would stop rebroadcasting the RREQ and sends a RREP to the source node with the complete route information consisting of the cached route in itself and the accumulated route record in the RREQ. In this way, the route may be established much quicker and the total delivery time and energy consumption are saved.

III. TTL Sequence-Based Expanding Ring Search

The goal of the expanding ring search is to find nodes that have the required route information to the destination node in their route cache by propagating RREQs. The propagation of RREQs is, on one hand, an efficient way for route discovery. It, on the other, may lead to ineffective flooding. Nodes in MANETs, for example, can be trapped in a loop of actions and end up asking each other for routing information for a long time without any solutions.

To control the flooding in MANETs, TTL (Time To Live) [1] sequence-based Expanding Ring Search is frequently used. The TTL sequence-based ERS restrains its searching range by giving RREQs a pre-defined TTL number. The TTL number corresponds to the radium of a searching area. Each time it fails to find any node that has route information to the destination node or the destination node itself, the source node rebroadcasts the RREQ in the next round with an increased TTL number to allow the RREQ to reach the remote nodes in further distance.

The TTL number increases linearly with a specified value [5]. The incremental value of TTL is fixed to 2 in [3] and [6], 1 in [7]. Figure 4 shows how TTL controls the RREQ relay range by predefined TTL values of 1, 2 and 3.

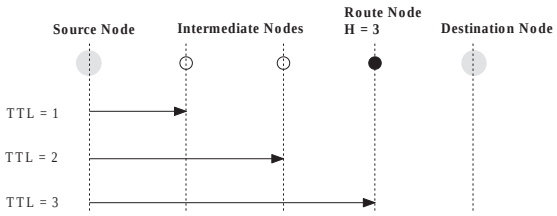


Fig. 4. TTL sequence-based ERS

Historically, TTL sequence-based mechanism was broadly adopted to reach all nodes in the networks to ensure successful route discovery in one round of flooding. Later an optimally chosen set of TTL was introduced to solve the generic minimal cost flooding search problem. However, it has been shown that there is no significant advantage from the optimal TTL sequence-based route discovery compared to the basic route discovery mechanism in terms of route discovery enhancement [8]. In addition, optimal TTL sequence-based discovery causes longer delay than the basic route discovery mechanism.

As it can be seen from Figure 4, in TTL sequence-based ERS, the source node issues and rebroadcasts a RREQ with increased TTL number if no RREP is received after a certain amount of time. The route discovery procedure repeats until the whole network is covered or until it finds the nodes that have the route information to the destination node or the destination node itself. This can also cause the overheads over the network area. More energy will inevitably be consumed during the route discovery if the route discovery process starts from the source node every time. It is especially costly when a larger area of the network needs to be searched.

IV. Blocking-ERS

We propose an alternative ERS scheme to support reactive protocols such as DRS and AODV, and it is called *Blocking Expanding Ring Search* (*Blocking-ERS* for short).

The Blocking-ERS integrates, instead of TTL sequences, a newly adopted control packet, *stop_instruction* and a hop number (H) to reduce the energy consumption during route discovery stage. We derive new route discovery algorithms (Algorithm 1, 2 and 3 for the source, intermediate and destination node respectively, and Algorithm 4 for blocking procedure).

The basic route discovery structure of Blocking-ERS is similar to that of conventional TTL sequence-based ERS. One of the differences from TTL sequence-based ERS is that the Blocking-ERS does not resume its route search procedure from the source node every time when a rebroadcast is required. The

rebroadcast can be initialised by any appropriate intermediate nodes. An intermediate node that performs a rebroadcast on behalf of the source node acts as a *relay* or an *agent* node. Figure 5 shows an example of our Blocking ERS approach in which the rebroadcasts are initialised by and begins from a relay node M in rebroadcast round 2, and another relay node N in round 3, and so on.

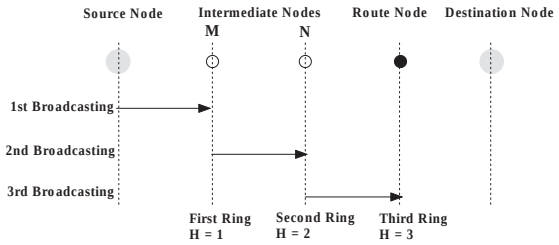


Fig. 5. Blocking-ERS

In Figure 5, the source node broadcasts a RREQ including a hop number (H) with an initial value of 1. Suppose that a neighbour M receives the RREQ with H=1, and the first ring was made. If no route node is found, that is, no node has the requested route information to the destination node, the nodes in the first ring rebroadcast the RREQ with an increased hop number, for example, RREQ with H=2 is rebroadcast in this case. The ring is expanded once again just like the normal expanding ring search in DSR [1] or AODV [9] (see also Figure 3), except with an extended waiting time.

We define the waiting time as

$$WaitingTime = 2 \times HopNumber$$

The nodes in Blocking-ERS receiving RREQs need to wait for a period of $2H$, i.e. $2\times$ their hop-number *unit time* before they decide to rebroadcast, where the 'unit time' is the amount of time taken for a packet to be delivered from one node to one-hop neighboring node.

In the example (Figure 5), the second ring has been made because there is no route node is found in the first ring. The node N receives RREQ and waits for, in this case, a period of 4 unit time since the hop number of the RREQ packet that the node N received is 2.

Two 'stop' signals are used in Blocking-ERS to control the flooding. One is the RREP, which can be sent by any route node to the source. The other is called 'stop_instruction' which can only be sent by the source node. The RREP informs the source node that 'a route node has been found' and therefore 'flooding should be stopped', while the 'stop_instruction' is an order to everyone involved in the flooding to terminate the route discovery process.

If there is no 'stop_instruction' message after 4 unit time has passed, it means that no node has the route information to the destination node in the second ring either. Then, every node in the second ring rebroadcasts to make the third ring. If a node finds the route information to the destination node in its route cache, this intermediate node is a route node, and

Algorithm 1 Source node

1: broadcast 'RREQ, $H = 1$ and max_j'
2: wait until a RREP is received
3: broadcast the 'stop_instruction' and H to everyone within the ring where it sent following conventional TTL scheme
4: use the 1st RREP for the data packet and save 2nd RREP as a backup
5: drop any later RREPs

Algorithm 2 Intermediate node

```
1: listen to RREQ
2: check the max_j after receiving the RREQ
3: if the H is bigger than the max_j then
4:    drop the RREQ
5: else
6:    check the route cache after receiving the RREQ
7:    if there is route information in the cache (i.e. being the
      Route Node) then
8:       send a RREP and H to the source node
9:    else
10:      wait for a period of 'waiting time'
         (i.e. 2 × HopNumber)
11:      while waiting do
12:         if receive a 'stop_instruction' then
13:            call the blocking procedure (see Algorithm 4)
14:            erase the source-destination pair in the route
               cache
15:         else if receives a 'RREP' then
16:            forward it to the source node
17:         end if
18:      end while
19:      if receives no 'stop_instruction' then
20:         increase the hop serial number by 1 and rebroad-
            cast RREQ
21:      end if
22:   end if
23: end if
```

it broadcasts the RREP message to the source node, and the source node broadcasts a 'stop_instruction' message to all the nodes involved in flooding. To improve the time efficiency, a 'stop_instruction' is sent via conventional TTL scheme. The automatic flooding takes place until the 'stop_instruction' message reaches all the nodes that have the same hop number as the route node that originated the RREP in the first place.

To limit the flooding in Blocking-ERS in case there is no route node is found, the source node sends a maximum journey value (max_j) with a RREQ packet. When an intermediate node receives a RREQ, the node compares the maximum journey value of the RREQ with its H. If the H is bigger than the max_j, then the intermediate node drops the RREQ packet.

To develop these ideas further, we have derived four al-

Algorithm 3 Route or destination node

1: wait for the first arriving RREQ
2: **if** receive a RREQ **then**
3: send the RREP and the H to the source route (contained in the RREQ packet)
4: **end if**

Algorithm 4 Procedure of blocking

1: compare H_r with H
2: **if** H_r is bigger **then**
3: forward the stop_instruction and
4: erase the source-destination pair in the route cache
5: **else**
6: drop the stop_instruction and
7: stop rebroadcasting and
8: erase the source-destination pair in the route cache
9: **end if**

gorithms Algorithm 1, 2 and 3 for the source, intermediate and destination (or route) nodes respectively, and Algorithm 4 gives the details of the blocking procedure.

The source node in Algorithm 1 covers the actions of a source node for the route discovery process. This includes initialising a route discovery process by first sending a RREQ (line 1), sending a 'stop_instruction' after a RREP is received (line 3) and handling the route information in RREPs (line 5, 6).

Similarly, Algorithm 2 summerises the actions taken by intermediate nodes which can be classified into route nodes and the rest. Once the route node is identified, a RREP will be sent with the current hop number to the source node (line 7, 8). Other intermediate nodes need to wait for a period of 'waiting time' (line 11–18) and start flooding if no 'stop_instruction' is received (line 19–21). During the waiting-time period, the intermediate nodes need to forward a 'stop_instruction' (line 11–12) or the RREP (line 8–10) because it might be the 2nd RREP for the source node as a backup.

Finally, Algorithm 3 highlights the actions by a route or destination node. It simply completes route information and sends it together with the RREP to the source (line 2–4).

V. Results

The Blocking-ERS proposed in this paper takes an approach of controlled flooding and continues its route discovery process from where it was left in the previous round each time after it fails to find a Route node (see Figure 5). This avoids the repeated network-wide flooding in the conventional TTL sequence-based ERS.

The source node broadcasts the next RREQ in the following round with an increased TTL number after the current route discovery is failed in TTL sequence-based ERS. Many intermediate nodes on the partial route that has been established so far are abandoned and the route needs to be rebuilt over again from the source node. The nodes may receive redundant RREQs repeatedly. Whenever no route to the destination node is found, the route discovery process starts from the very beginning (see Figure 4).

A clear contrast can be seen from Figure 4 and 5 and comparisons can be made between the proposed Blocking-ERS and the conventional TTL sequence-based ERS.

A. Energy Consumption

Energy consumption during the transmission of RREQs can be saved by using the Blocking-ERS scheme.

Let the amount of energy consumption on each node for one broadcast be the same *unit energy* consumption, denoted by *UnitEnergy*. We assume that each action of broadcasting a RREP, RREQ or 'stop_instruction' consumes the same amount of 1 UnitEnergy.

This can be easily shown by the difference of the energy consumption between the conventional TTL sequence-based ERS and the Blocking-ERS scheme.

1) One route case: We first consider only the energy consumption along the route from the source to the route node (Figure 6).

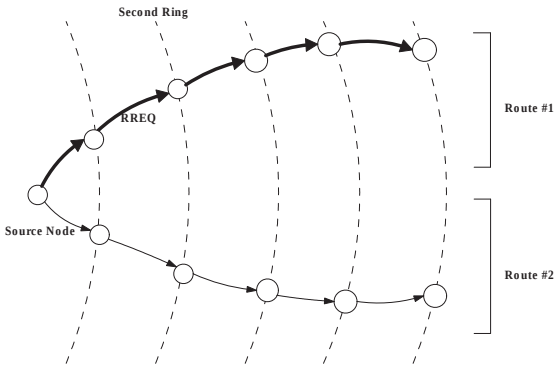


Fig. 6. Energy consumption for one route

The energy consumption for the TTL based ERS and for the Blocking-ERS can be described by the following formula respectively, where H_r is the hop number of the route node:

$$E_{TTL-ERS} = H_r + \sum_{i=1}^{H_r} i \text{ (UnitEnergy)}$$

$$E_{Blocking-ERS} = 3H_r \text{ (UnitEnergy)}$$

The difference of the amount of energy consumption is more visible from Figure 7, where the amount of energy consumption by the two ERS approaches are plotted against the number of rings (i.e. the searching distance between the source to the route node). The Blocking-ERS curve is below the TTL based ERS curve after ring 1. As we can see, the difference of the amount of energy consumption between these two mechanisms becomes larger as the distance increases between the source and the route node.

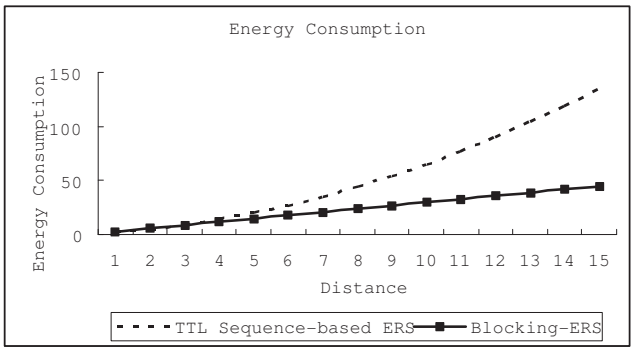


Fig. 7. Comparison of Energy Consumption for One Route

2) General case: We now consider the general case.

For the Blocking-ERS, the energy consumption during the route discovery process can be considered as the total energy consumption in three stages: (a) searching for the route node, (b) retrun RREP and (c) sending the 'stop_instruction'.

For the conventional TTL based ERS, the energy consumption during the route discovery process includes that in two stages: (a) searching for the route node, and (b) return RREP.

The energy consumed for '(b) returning a RREP' is H_r UnitEnergy for both routing schemes, and H_r UnitEnergy is consumed for the Blocking-ERS stage '(c) sending the stop_instruction'.

In the stage of '(a) searching for the route node', the energy consumption is different for the two methods.

Each ring contains a number of nodes that rebroadcast to form the next ring. Let n_i be the number of nodes in ring i and the hop number of the route node be H_r (see Figure 6 and 8).

In the Blocking-ERS, the energy consumed in each ring is as blow:

Ring i	Energy Consumed
0	1
1	n_1
2	n_2
$\vdots$	
$H_r - 1$	n_{H_r-1}

In the TTL based ERS, the energy consumed in each ring is as follows:

Ring i	Energy Consumed
0	1
1	$1 + n_1$
2	$1 + n_1 + n_2$
$\vdots$	
$H_r - 1$	$1 + n_1 + n_2 + \cdots + n_{H_r-1}$

Therefore, the total energy consumption by the Blocking-ERS is

$$E_{B-ERS} = 2(1 + \sum_{i=1}^{H_r-1} n_i) + E_{RREP} \text{ (UnitEnergy)}$$

Similiarly, the total energy consumption by the conventional TTL sequence based ERS is

$$E_{TTL\_ERS} = H_r + \sum_{i=1}^{H_r-1} \sum_{j=1}^{i} n_j + E_{RREP} \text{ (UnitEnergy)}$$

The difference between the E_{B_ERS} and E_{TTL_ERS} is

$$E_{saved} = H_r - 2 + \sum_{i=1}^{H_r-1} ((\sum_{j=1}^{i} n_j) - 2n_i) \text{ (UnitEnergy)}$$

Clearly, when $n_i = 1$, for $i = 1, \cdots, H_r$. The above formula represent the energy consumption for a single route. This indicates that the energy consumption saving achieved by the Blocking-ERS for a single route is the *minimum* amount of energy saving. In reality, with a high possibility that more than one route are involved with the flooding (Figire 8). More energy saving can, therefore, be achived during the route discovery period.

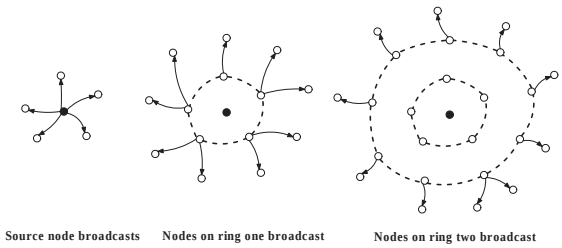


Fig. 8. Broadcasting nodes in each ring in Blocking-ERS

B. Time Delay

We consider the *time delay* for the route discovery period, during which from the RREQ is broadcasted for the first time, transmitted from the source node to the route node possibly via flooding. That is the total time taken from when the source node broadcasts the first RREQ for the first time until after a Route Node is found and the source node receive the RREP from the Route Node.

Let the *UnitTime* be the one-hop transmission time, which is the time taken for a RREQ from a broadcasting node to one of its neighbour nodes.

In case of TTL sequence based ERS, in Figure 4, for example, suppose $H = 3$, that is, the route node is 3 hops distant from the source node. The total time includes the time when $TTL = 1, 2$ and 3. The final TTL number equals to the hop number of the route node. This gives the following formula of total time delay for the TTL sequence-based ERS:

$$T_{TTL\_ERS} = 2\sum_{i=1}^{H_r} i \text{ (UnitTime)}$$

Now consider the time delay in the Blocking-ERS. The total time includes the time for three stages: (a) searching for the

route node, (b) returning the RREP and (c) broadcasting the 'stop_instruction'. For stage (a), the time consists of the time to for broadcasting and the waiting time. The broadcasting time for 1 hop distance is 1 UnitTime. The waiting time depends on the hop number of the node. In Figure 5, for example, the route node is 3 hops distant from the source node. Each node needs to wait for $2H$ before rebroadcasting. At ring 1, the node waits for $2 \times 1 = 2$ UnitTime, and at ring 2, the node waits for $2 \times 2 = 4$ UnitTime, and at ring 3, the node waits for $2 \times 3 = 6$, so the total waiting time for the '(a) stage of searching for the route node' is $2 + 4 + 6 = 12$, and the total time for stage (a) is $12 + H_r = 12 + 3 = 15$. The time for stage (b) and (c) are H_r and this gives $2H_r = 2 \times 3 = 6$. Therefore, the total time for the route discovery and flooding control is $15 + 6 = 21$ UnitTime. Mathematical formula is presented below, where H represents the hop number of a route node.

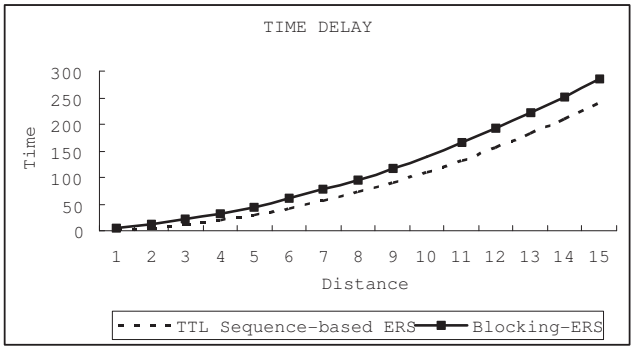


Fig. 9. Comparison of the time delay

The formula of the time delay in the Blocking-ERS is given

$$T_{B-ERS} = 3H_r + 2\sum_{i=1}^{H_r} i \text{ (UnitTime)}$$

Compare this to the TTL sequence based ERS:

$$T_{TTL\_ERS} = 2\sum_{i=1}^{H_r} i \text{ (UnitTime)}$$

It is clear that the difference between the two is $3H_r$, three times of the hop serial number of the route node, depending only on the distance between the source node to the Route node.

The time difference is caused by the fact that it is the intermediate nodes, instead of the source node, that wait for the 'stop_instruction' in the Blocking-ERS. If there is no 'stop_instruction after a certain amount of time, the intermediate node received RREQ recently will rebroadcasts RREQ automatically to continue the route discovery procedure.

The time delay in both approaches is compared in Figure 9. As illustrated, the Blocking-ERS takes slightly more time than the conventional TTL sequence-based ERS for the route discovery process.

VI. Conclusion

We have introduced a new method 'Blocking-ERS' for route discovery. The analysis demonstrates a substantial improvement in energy consumption that can be achieved by the Blocking-ERS at a margin cost of slightly longer time. The amount of the energy saving by the Blocking-ERS depends on the number of nodes involved in flooding and the number of searching rings in general case. The minimum energy saving achieved by the Blocking-ERS can be seen clearly from the one route case and even the minimum energy saving looks promising.

Acknowledgment

The authours wish to thank the anonymous referees for helpful comments, and to Seungwoo Yang and Alan Huang for useful discussion.

References

[1] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imlellnski and H. Korth, Eds. Kluwer, 1996, pp. 153–181.

[2] M. Stemm and R. H. Katz, "Measuring and reducing energy consumption of network interfaces in hand-held devices," *EICE (Institute of Electronics, Information and Communication Engineers) Transactions on Communications*, vol. E80-B(8), pp. 1125–1131, 1997.

[3] C. Perkins, E. Royer, and S. Das. (2003) Ad hoc on-demand distance vector (aodv) routing. IETF Request for Comment. [Online]. Available: www.ietf.org/rfc/rfc3561.txt

[4] B. Liang and Z. Haas, "Virtual backbone generation and maintenance in ad hoc network mobility management," in *In Proc. InfoCom 2000*, 2000, pp. 1293–1302.

[5] J. Hassan and S.Jha, "On the optimazation trade-offs of expanding ring search," in *Proc. of IWDC 2004*, 2004, pp. 489–494.

[6] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceeding of the ACM Sigmetrics Conference*, Mariana del Rey,CA, 2002.

[7] E. Royer, "Routing in ad-hoc mobile networks: On-demand and hierarchical strategies," Ph.D. dissertation, University of California at Santa Barbara, 2000.

[8] D. Koutsonikolas, S. Das, H. Pucha, and Y. C. Hu., "On optimal ttl sequence-based route discovery in manets," in *Proc. of the 2nd ICDCS International Workshop on Wireless Ad Hoc Networking (IEEE WWAN 2005)*, Columbus, Ohio, 2005.

[9] C. Perkins and E. Royer, "Ad -hoc on-demand distance vector routing," in *The 2nd Annual IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, 1999, pp. 90–100.

A New Virtual Backbone for Wireless Ad-Hoc Sensor Networks with Connected Dominating Set

Reza Azarderakhsh, Amir H. Jahangir and Manijeh Keshtgary*
Department of Computer Engineering
Sharif University of Technology, Tehran,Iran
azarderakhsh@ce.sharif.ac.ir, jahangir@sina.sharif.ac.ir, keshtgar@mehr.sharif.ac.ir

Abstract

A wireless Ad-hoc sensor network consists of a number of sensors spread across a geographical area as a collection of sensors that form an ad-hoc wireless network. Sensors are very tiny devices that their primary function is to sense the target, convert the signal into a suitable data format, and pass on the data to a command node. These sensor nodes are very heavily constrained in processing power, and have a limited energy supply. Since energy is such a scarce resource, several algorithms have been developed at the routing and MAC layers to utilize energy efficiently and extend the lifetime of the network. First layer of the sensor networks is the infrastructure layer and there is no backbone for these networks. In this paper, we propose a virtual backbone for these networks and we measure the network lifetime and survivability as the performance evaluation metrics of the proposed model.

1. Introduction

Sensor networks have attracted recent research attention due to wide range of applications in civil and military settings [9]. The sensors can be deployed in large numbers in wide geographical areas and can be used to monitor, detect and report time-critical events like earthquakes, chemical spills, or the position of moving objects [1]. These sensors are typically disposable and are expected to last until their energy drains. Their small size factor limits their processing and communication abilities. Energy is the scarcest resource for such nodes, and it has to be managed wisely to extend the lifetime of a sensor for the duration of the mission. Sensors are equipped with data processing and communication abilities. The former consists of a sensing circuit that captures data from the target environment and converts them into an electrical signal. The signal is then transmitted via radio transmitter to a command center either directly or through a data concentration center (gateway). The gateway can perform fusion of data received from the sensors and also filtering out the erroneous data, before passing it on to the command center. These data processing and communication activities are the main consumers of a sensor's energy. A battery-operated sensor cannot be kept active all the time since this will deplete its energy resources quickly. The sensor should be notified on when exactly to turn on its circuitry for performing various functions like sensing, transmitting and receiving data. This is done at the MAC layer by a TDMA scheme, in which each sensor is allotted a time slot for performing its duties [9, 7] or other routing like span [3]. The sensor turns off its circuitry when its time slot has passed to conserve energy. A properly designed slot allocation scheme can extend the lifetime of sensors. The energy consumed for transmitting data increases with the distance between the communicating parties. Thus, the sensors that are located at longer distances from the gateway or command center will die out more quickly than those at shorter distances. Instead of having the longer distances sensors transmitting directly to the gateway, certain sensors in between can be used as the forwarding agents. Transmitting data over these much shorter distances leads to substantial power savings at sensors. This approach is at the routing layer, and requires efficient algorithms to dynamically establish routes between sensor nodes. Using connected dominating set in wireless sensor network is a new method for saving energy consumption.

The rest of the paper is structured as follows. We next describe preliminaries on connected dominating set. Section 3 describes the related work. We present our proposed algorithm in section 4. Simulation results are used to evaluate the model in Section 5. Finally Section 6 concludes the paper.

* Instuctor at Shiraz University of Technology

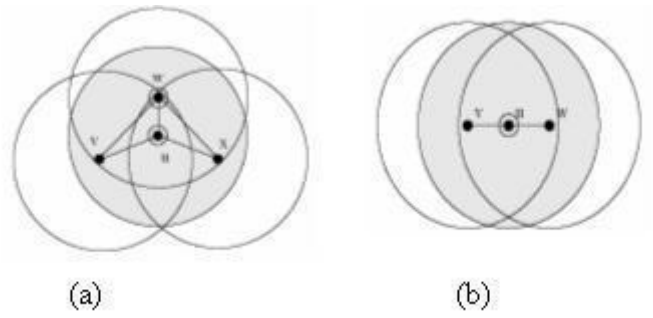


Figure 1: Example of Ad-hoc network

2. Preliminaries

An ad-hoc network can be presented by a unit disk graph where every vertex (host) is associated with a disk connected at this vertex with the same radius (transmitter range) [4]. Two vertices are neighbors if and only if they are covered by each others disk. For example, both vertices v and w in Figure 1(b) are neighbors of vertex u because they are covered by disk u, while vertices v and x in Figure 1(a) are not neighbors. In an Ad-hoc network, some links (edges) may be unidirectional due to the disparity of energy levels of hosts. Therefore a general Ad-hoc wireless sensor network can be considered as a directed graph with a high percentage of bidirectional links.

Routing in wireless Ad-hoc sensor networks is harder than in wired networks, due to the limited resource, network mobility, and lack of physical infrastructure. Virtual infrastructures, such as a Connected Dominating Set (CDS), were proposed to reduce the routing overhead and enhance scalability [5]. In dominating set, every vertex in the graph is either in the set or adjacent to a vertex in the set. Vertices in dominating set are also called gateways while vertices that are outside a dominating set are called non-gateways. Among CDS based routing protocols, only gateways need to keep routing information in a proactive approach and the search space is reduced to the dominating set in a reactive approach [10].

Unfortunately, finding a minimum connected dominating set is NP-complete for most graphs. Wu and Li proposed a simple and efficient approach called marking process which can quickly determine a CDS [10]. This approach was first proposed for undirected graphs using the notion of dominating set only and was later extended over directed graphs by introducing another notion called absorbent [11]. Specifically, each host is marked if it has two unconnected neighbors. It is shown that collectively these hosts achieve a desired global objective - a set of marked hosts forms a small CDS. Based on the marking process, vertices u and w in Figure 1(b) are marked and they form a dominating set in their network. The CDS derived from the marking process is further reduced by applying two dominant pruning rules. According to dominant pruning rule 1, a marked host can unmark itself if its neighbor set is covered by another marked host; that is if all its neighbors are connected with each other via another gateway, it can relinquish its responsibility as a gateway. In Figure 1(b), either u or w can be unmarked (but not both). According to rule 2, a marked host can unmark itself if its neighborhood is covered by two other directly connected marked host. The marking process 1 and 2 are purely localized algorithm where the marker of host depends on topology of small vicinity only [10].

We propose a connected dominating and a clustering algorithm in the MAC layer that can increase the lifetime and survivability of the wireless sensor network.

3. Related Work

Das et al. [6] proposed an algorithm to identify a sub network that forms a minimum CDS (MCDS). This algorithm finds a CDS by growing a tree T starting a vertex with the maximum vertex degree, and adding a new vertex to T according to its effective degree (number of neighbors that are not neighbors of T). The main drawback of this algorithm is its centralized style. The mesh scheme [2] designates a subset of border members as gateways so that there is exactly one virtual link between two neighboring clusters.

Wu and Dai's [10] marking process uses a constant number of rounds to determine a CDS. This approach can be applied to Ad-hoc sensor networks with bidirectional link only. In this paper, we introduce a model to develop a virtual infrastructure for wireless senor networks. Our algorithm consists of 2 phases: First, we cluster sensor nodes using clustering algorithm and then we implement the CDS algorithm to intra clusters.

4. Our Proposed Method

We propose a way to making a virtual infrastructure for wireless sensor network with connected dominating set and its optimization methods in addition to a clustering method for sensor nodes. The wireless sensor nodes are densely deployed in the space. Therefore, if we can cluster them with their gateways (cluster head), the network life time and survivability increases and the routing algorithm will be easy [8, 9]. Lifetime of a sensor network is defined as the time after which certain fraction of sensor nodes run out of their batteries, resulting in a routing hole within the network. The sensor network is divided into separate regions known as clusters. A cluster is nothing but a gateway and a set of sensors that communicate with that gateway. Dividing a network into clusters has the advantage of reducing routing overhead. Since there are fewer sensor nodes in a cluster, the computation of routes is much faster. The number of

clusters in the network is equal to the number of gateways, since there is one gateway for each cluster. After clustering the network into a subset of clusters with the proposed algorithm, then we use the CDS finding algorithm. In the simulation, we show that the network lifetime and packet overhead are optimized. We use the packet drop probability as the performance metrics parameter.

4.1. Clustering Algorithm

The main objective of our approach is to cluster sensor network efficiently around few high-energy gateway nodes. The clustering algorithm is responsible for dividing the whole network topology into clusters. The algorithm takes the set of sensor nodes and gateways, and partitions them such that there is one gateway and a subset of the set of nodes in each cluster. Clustering enables network scalability to large number of sensors and extends the life of the network by allowing the sensors to conserve energy through communication with closer nodes and by balancing the load among the gateway nodes. Gateways associate cost to communicate with each sensor in the network. Clusters are formed based on the cost of communication and the load on the gateways. Network setup is performed in two stages; 'Bootstrapping' and 'Clustering'. In the bootstrapping phase, gateways discover the nodes that are located within their communication range. Gateways broadcast a message indicating the start of clustering. We assume that receivers of sensors are open throughout the clustering process. Each gateway starts the clustering at a different instance of time in order to avoid collisions. In reply the sensors also broadcast a message with their maximum transmission power indicating their location and energy reserve in this message. Each node discovered in this phase is included in a range set per gateway. Bootstrapping a sensor network is the processing of establishing inter-node links and forming an overall network topology. Bootstrapping typically consists of two phases:

- Node Discovery: unless the nodes are manually placed, nodes are not aware of their peers and thus should at least discover their neighbors.
- Topology Setup: Based on the established links among neighboring nodes, a network topology should be established to allow for data gathering.

In the clustering phase, gateways calculate the cost of communication with each node in the range set. This information is then exchanged between all the gateways. After receiving the data from all the other gateways each gateway start clustering nodes based on the communication cost and the current load on its cluster. When the clustering is over, all the sensors are informed about the ID of the cluster

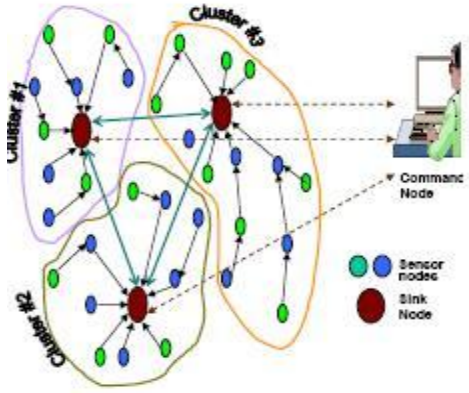


Figure 2: Clustering Nodes in sensor network

they belong to. Since gateways share the common information during clustering, each sensor belongs to only one cluster. For inter-cluster communication all the traffic is routed through the gateways. The clustering algorithm may use a number of metrics to determine how to form clusters [12]:

- Physical distance of a sensor node from the sensor. In this case, the sensor node determines the nearest gateway and reports to that gateway.
- Equal number of sensors in each cluster. This ensures that each gateway has equal routing overhead.
- Redundancy assurance. A sensor may determine that there exists more than one gateway within transmission range. It chooses one gateway and joins that cluster, and keeps the others as backups. Thus, this algorithm may change the cluster formation due to factors such as gateway failure.

As shown in Figure 2, at first, sensor nodes are clustered into clusters and then the CDS algorithm is applied to the clustered network. The algorithm that has been used in our approach uses the proximity-based metrics. Thus, each sensor node chooses the nearest gateway and joins that cluster. As shown in Figure 2, all sensors find their Euclidian distance from the gateways and join them. After clustering sensor nodes in clusters with cluster head or gateways, the CDS finding algorithm is used in each cluster.

4.2. CDS Algorithm

An Ad-hoc wireless sensor network with unidirectional links can be represented by a simple directed graph $G = (V, E)$, where V is a set of vertices (hosts) and E is a set of directed edges (unidirectional links). A directed edge directed from u to v is denoted by an ordered pair (u,v). A directed graph G is strongly connected if for any two vertices u and v , a (u,v)-path, i.e., a path connecting u to v, exists. If (u, v) is an edge in G, we say that u dominates v, and

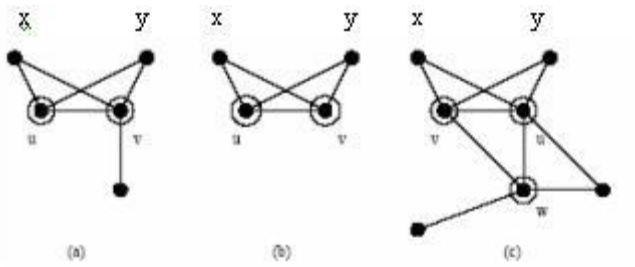


Figure 3: Dominating Set Reduction examples

v is the absorbent of u. The dominating neighbor set Nd (u) of vertex is defined as

$$Nd(u) = \{w : (w, u)\epsilon E\} \quad (1)$$

The absorbent neighbor set Na(u) of vertex u is defined as

$$Na(u) = \{v : (u, v)\epsilon E\} \quad (2)$$

and N(u) represents the neighbor set of vertex u:

$$N(u) = Nd(u)UNa(u) \quad (3)$$

Suppose the marking process is applied to the network that is redrawn in Figure 3(a). Host u will be marked because (x,u) belongs to E and (u,y) belongs to E, but (x,y) doesn't belong to E. Host v will also be marked. All other hosts will remain unmarked because no such pair of neighbor hosts can be found. The marking process is as below:

Algorithm 1: Marking Process[2]

1. Each u periodically exchanges its neighbor set Nd(u) and Na(u) with all its neighbors.
2. u sets its marker to T (marked) if there exist two neighbors v and w of u such that (w,v) belongs to E, and (u,v) belongs to E and (w,v) does not belong to E.

Initially vertices are unmarked. They exchange their open neighborhood information with their one-hop neighbors. Therefore each node knows all of its two-hop neighbors. The marking process uses the following simple rule: any vertex having two unconnected neighbors is marked as a dominator. The set of marked vertices form a connected dominating set, with lots of redundant nodes. Two pruning rules are provided to post-process the dominating set, based on the neighborhood subset coverage. A node u can be taken out from S, the CDS, if there exists a node v with higher ID such that the closed neighbor set of u is a subset of the closed neighbor set of v. For the same reason, a node u will be deleted from S when two of its connected neighbor in S with higher IDs can cover all u's neighbors. This idea is also extended to directed graph. Due to differences in transmission ranges of wireless networks, some links in Ad-hoc wireless network may be unidirectional. In order to apply this to a directed graph like sensor network model, neighboring vertices of a certain node are classified into a dominating neighbor set and an absorbent neighbor sets in terms of the directions of the connected edges.

5. Simulations

The simulation has two different parts. One is the Connected Dominating Set algorithm used by Ji and Dai in [2] to simulate the connected dominating set reduction in Ad-hoc wireless sensor networks and we extend it by adding the clustering algorithm to it. The second part is a general wireless sensor network simulator. To generate a random Ad-hoc network, n hosts are randomly placed in a restricted 100*100 (meter)area. The transmitter range R is adjusted according to the average vertex degree d to produce nd/2 links in the corresponding unit disk graph. Most of these links are treated as bidirectional, but a small portion (p%) of them are randomly selected to be unidirectional links. Networks that cannot form a strongly connected graph are discarded. For each combination of parameters (n, d and p), the simulations repeated 500-2000 times until the confidence interval is sufficiency small (+/-1% for the confidence level of 90%).

For simulating the clustering of sensor nodes in the network, we make a network with below entities: Sensor nodes, Gateways, Clusters, Packets, Packet Queues, Targets, User-interface Events, Event Queues. This calls for an object-oriented design approach to the simulator. Each entity is modeled by a separate object that encapsulates its functionality. These objects represent a high-level decomposition of the sensor network allowing us to establish the interactions between the entities. Our metrics are the energy consumption per node, the average energy consumption within overall network, and network lifetime [6]. The average energy consumption for a uniformly 10-cluster network is less than 50uj, as can be seen in Figure 4. In Figure 5, we can see that fewer than 30 % of the sensors consumed more than 50uj. The density of gateways can be increased to reduce the average energy consumption.

6. Conclusions and Future Work

We have presented a virtual infrastructure backbone for wireless sensor network with the concept of connected dominated graph and we measure the network energy consumption as an important parameter to evaluate network lifetime. To our knowledge this is the first time that a sensor network introduced with clustering and infrastructure

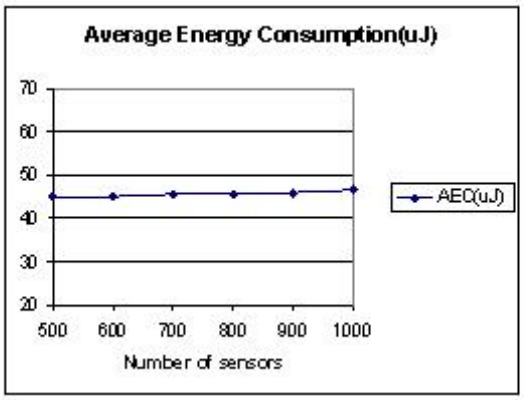


Figure 4: The average energy consumed by sensors in a network with 10 clusters.

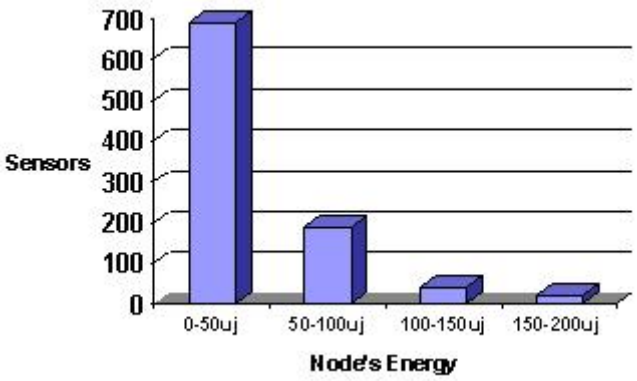


Figure 5: Distribution of sensor energy consumption with our approach

concepts together. In our simulation, we can decompose the wireless sensor network in an object oriented way and we can measure other parameters like Packet Drop Probability (PDP) as a network survivability metrics for our future work. We will also focus on evaluating survivability of the sensor networks considering PDP and network lifetime parameters.

References

[1] Akyldiz, I., Su, W., Sankarasubramaniam, Y. and Cayiroici, E. ” A survey on Sensor Network, ” IEEE Communications Magazine, vol. 40, Issue:8, pp. 102-114,August 2002.

[2] Arisha, K., Youssef, M. and Younis, M., ”Energy-Aware TDMA-Based MAC for Sensor Networks,” in the Proceedings of the IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002), New York City, New York, May 2002.

[3] Chen, B., Jamieson, K., Balakrishnan H. and Morris, R. ”Span: an Energy-Efficient Coordination Algorithm for Topology Maintenance,” in Proc.of the 7th Annual International Conf. on Mobile Computing and Networking, July 2001.

[4] Clerk, B. N. and Colbourn, C. J. ” Unit Disk Graph ,” Discrete Mathematics, Vol. 86, pp. 165-177, 1990.

[5] Dai, F. and Wu, J. ”‘An extended localized algorithm for connected dominating set formation in ad hoc wireless networks”’, IEEE Transactions on Parallel and Distributed Systems, 15(10):908-920, Oct. 2004.

[6] Das, B. and Sivakumar, R. ”Routing in Ad-hoc networks using a spine,” Proc. Of ICCN, PP 1-20, Sept 1999.

[7] Heinzelman, W., Kulik, J. and Bakakrishnan, H. ” Negotiation -based Protocols for Disseminating in Wireless Sensor Network,” in Proc. Of the 5th Annual ACM/IEEE International Conf. on Mobile Computing and Networking, 1999.

[8] Lindesy, S. and Raghavendra, C. ” PEGASIS : Power-Efficient Gathering in Sensor Information System,” International Conf.‘on Communications, 2001.

[9] Servetto, S. and Barrenechea, S. ”Constrained Random Walks on Random Graphs: Routing Algorithm for Large Scale Wireless Sensor Netwotks,” in Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia, USA, 2002.

[10] Wu, J. and Dai, F. ”‘ A distributed formation of a virtual backbone in manets using adjustable transmission ranges”’, In Proc. of ICDCS, pages 372-379, Mar. 2004.

[11] Younis, M., Youssef, M. and Arisha, K. ”Energy-Aware Routing in Cluster-Based Sensor Networks”, in the Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002), Fort Worth, Texas, October 2002.

[12] Younis, M., Youssef, M. and Arisha, K. ”Energy-Aware management in Cluster-Based Sensor Networks,” The International Journal on Computer Networks, Vol. 43, No. 5, pp. 649-668, December 2003.

QoS Preserving Topology Advertising Reduction for OLSR Routing Protocol for Mobile Ad Hoc Networks

Luminita Moraru and David Simplot-Ryl
IRCICA / LIFL, University of Lille 1, France
INRIA futurs, POPS research group
Email : {Luminita.Moraru, David.Simplot}@lifl.fr

Abstract— **Mobile ad hoc networks (MANET) are formed by mobile nodes with a limited communication range. Routing protocols use a best effort strategy to select the path between a source and a destination. Recently, mobile ad hoc networks are facing a new challenge, quality of service (QoS) routing. QoS is concerned with choosing paths that provide the required performances, specified mainly in terms of the bandwidth and the delay. In this paper we propose a QoS routing protocol. Each node forwards messages to their destination based on the information received during periodically broadcasts. It uses two different sets of neighbors: one to forward QoS compliant application messages and another to disseminate local information about the network. The former is built based on 2-hop information knowledge about the metric imposed by the QoS. The latter is selected in order to minimize the number of sent broadcasts. We provide simulation results to compare the performances with similar QoS protocols.**

I. Introduction

In the context of mobile ad hoc networks[1], new challenges are raised for routing protocols. Nodes are communicating through wireless links with limited range. Each message sent by a node will be received only by the nodes located in this communication range. Additionally, links between nodes are not stable due to the nodes mobility.

Routing protocols are finding paths between a source and a destination that do not communicate directly. They consider the number of hops as criterion for finding optimal routes between nodes. In the case of QoS routing [2], new constraints become prioritary (bandwidth, delay) and new metrics must be considered. When a packet coming from the application layer is routed to its destination, the links between nodes are relevant only if they are compliant with the QoS requirements. Many of the solutions that have been proposed to this problem are enhancements of existing routing protocols.

We consider the particular situation of proactive protocols, where each node stores routing tables with all known destinations in the network. Hosts are aware of network topology due to the routing related information, periodically propagated into the network. Each node sends periodically broadcasts about the links with its neighbors. Existing proactive protocols (e.g. OLSR [3]) minimize the number of broadcasts by selecting only a subset of neighbors, multipoint relays (MPR) [4], to relay messages containing routing related information. The MPR set of a node is computed between direct neighbors, by a greedy heuristic, to cover all neighbors at a distance of 2 hops. The same set of nodes is used for packets routing.

When guaranteed QoS is demanded, an option is to modify existing protocols to use only the links respecting QoS requirements. This will impose additional conditions to the neighbors subset selected as relays, thus the number of selected neighbors and the network traffic are increased.

This paper presents a method for QoS paths selection, based on network topology complexity reduction. Only the neighbors that are providing maximum bandwidth links are advertised. In our solution, we determine the 1-hop neighbors representing the best paths to the set of 2-hop neighbors, in terms of a specific metric. First we eliminate from redundant paths, the worst performance link. Since each node has complete knowledge only until the 2 hop distance neighbors, redundant paths are represented by nodes that are both 1-hop and 2-hop neighbors. Then, we are making the selection considering a specific QoS metric. By selecting only nodes providing optimal links, we are reducing the complexity of network topology, while preserving the connectivity of the network and the availability of paths. QoS enabled routing uses selected neighbors set when it forwards application messages. Therefore, the selection is flooded into the entire network. We use MPR sets to flood the selection of a node.

The paper is organized as follows: first a presentation of existing QoS protocols is made. Next section contains a description of OLSR protocol, for which we proposed an enhancement, followed by the description of the algorithm used for advertised set selection, for concave constraints (e.g. bandwidth) in section IV and for additive constraints (e.g delay) in section V. Experimental results are presented in section VI and conclusions in section VII.

II. Previous work

QoS routing protocols developed for mobile ad hoc networks [5] are extending classic, best effort routing algorithms for MANET.

On demand routing protocols are using different communication models in order to satisfy the QoS requirements, e.g. TDMA (Time Division Multiple Access) or CDMA (Code Division Multiple Access) over TDMA. The issues raised are bandwidth or delay calculation and resource reservation

during path discovery. An enhanced version of Ad-hoc On demand Distance Vector (AODV) protocol for QoS support [6] introduces a mechanism for resource reservation simultaneous with path discovery. An extension of Dynamic Source Routing (DSR) protocol is presented in [7]. It deals with common problems in TDMA environment for bandwidth reservation (e.g. race condition, parallel reservation problem). Temporally Ordered Routing Algorithm (TORA) extension [8] chooses from the available paths the shortest path compliant with the QoS requirements. The disadvantage is that they are operating not only into the network but also into Medium Access Control (MAC) layer.

From the reactive protocols category, an extension of OLSR for optimal routes in terms of QoS requirements was proposed in [9]. QOLSR proposes a heuristic for MPR selection and imposes several conditions for these nodes, in order to provide an optimal path, both in terms of hop distance and QoS metric. QOLSR has the disadvantage of increasing the number of MPR relays, thus the number of broadcasts in the network.

Another approach is core-extraction distributed ad hoc routing (CEDAR) protocol [10]. It determines a core dominating set. Only the nodes in this set are aware of core topology and of the metric of the neighbor links. This limits the number of broadcasts, compared with the control flooding of reactive protocols.

III. OLSR Protocol Adaptation

Optimized Link State Routing (OLSR) protocol is a table driven protocol for MANET.

It maintains tables containing all the necessary data for finding a path to any other node in the network. In order to keep up to date routes, it regularly propagates routing information. It uses two types of messages: *HELLO messages* for neighborhood discovery and *topology control (TC) messages* for entire network topology discovery. *HELLO messages* are advertising the neighbors and MPR sets, while *TC messages* are disseminating network topology information necessary for building routing tables. MPR sets are enough to compute best routing path.

By using different sets of nodes for routing and topology advertising, new data structures are added to the *information base* of each node. Similarly to OLSR each node stores the 1 and 2-hop neighbors, MPR and MPR selector sets. Additionally each node will maintain the QoS Advertised Neighbor Set (QANS), which provides optimal connectivity based on the imposed metric and a list of QANS selectors: neighbors that selected it in their QANS set.

Topology information maintained at each node is retrieved from the TC messages and contains the list of all know destinations in the network together with the list of the last hop used to reach them. In OLSR this list contains the links of a node with its MPR selectors. In our case, these links are replaced in the TC messages by the QANS selectors set. Each node that receives a TC message will broadcast it only if it is in the MPR list of the last sender of the message.

IV. Topology filtering for bandwidth

A. Graph density reduction

Bandwidth constraint routing is based on finding routes in a network that maximize this criterion. A node has at most information regarding the presence of 1-hop and 2-hop neighbors and the metric of all 1-hop neighbors links. Based on link metric each node reduces the broadcasted information only to information needed to compute paths with the respect to constraints.

We consider the model of a network represented by a graph $G = (V, E)$, where V is the set of vertices in the graph, associated to the network nodes and E is the set of edges, representing links between nodes. Each communication link is characterized by a bandwidth value. Let B be the value of the maximum bandwidth link in the network. Then, we can define b, the bandwidth function that maps the set of edges E to the interval $]0, B]$. If the links are bidirectional, function b is considered to be symmetric (*i.e.* $b(u, v) = b(v, u)$). Bandwidth is a concave constraint, the bandwidth of a path p is defined by the minimum bandwidth link on that path. This means that for $p = \{a_0, a_1, \ldots, a_n\}$, the bandwidth b_p of p is equal to:

$$b_p = \min_{0 \leq i < n} \{b(a_i, a_{i+1})\}.$$

We will present below the method used for reducing the density of the graph. It is based on the situation where a node n_2 is a common neighbor for both a node u and another 1-hop neighbor of u, n_1. A triangle is generated in the graph. This is often the case of networks represented by a dense graph. Each node will maintain locally two paths to both neighbors (*e.g.* between n_1 and n_2 there are $p_1 = \{n_1, n_2\}$ and $p_2 = \{n_1, u, n_2\}$), characterized by the bandwidths: b_{p_1} and b_{p_2}. We can reduce the density of the graph by eliminating from the triangle formed by u, n_1 and n_2 the link with the minimum bandwidth.

Fig. 1 represents an example. In 1(a), $b_{p_1} = 3$ and $b_{p_2} = 4$. This makes p_2 the preferred option when maximum bandwidth routes are necessary. Both (n_1, n_2) and (n_2, n_3) have redundant paths with better metric value, as shown in 1(b) and they are eliminated.

Let us define the graph $G' = (V', E')$ containing the remaining set of edges:

$$\begin{aligned} E' \quad = \quad & \{(u, v)\, in\, E |\ \nexists w\, such\, that\, (u, w), (v, w) \in E \\ & \wedge\, b(u, v) \leq \min(b(u, w), b(v, w))\}. \end{aligned}$$

This graph reduction is a variation of Relative Neighborhood Graph (RNG) [11].

For a weight function f, the RNG graph,$G_{RNG} = (V, E_{RNG})$ of G, imposes the following condition, for an edge $(u, v) \in E$ between vertices u and v to exists:

$$\forall w \in V, w \neq u\, and\, v, f(u, v) \geq \max(f(u, w), f(v, w)).$$

Similarly, for the bandwidth metric, G' will represents the initial graph reduced to the RNG, which uses the bandwidth as weight function instead of distance.

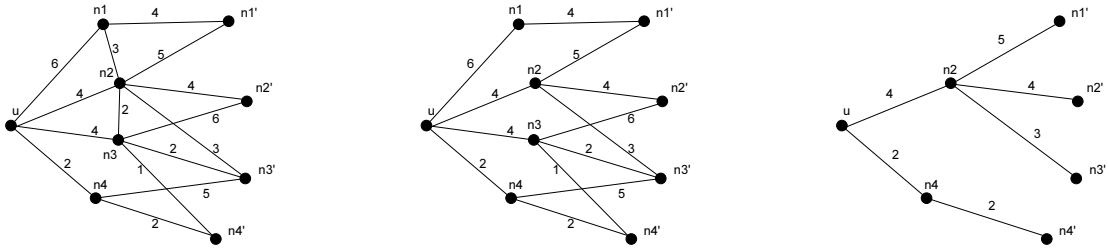


Fig. 1. Example of bandwidth QANS selection for a node

In the case of two equal minimum links, another two criteria are evaluated in order to choose the link that will be eliminated. They are based on nodes IDs comparison, since each node is identified by an ID, unique in the network. First, the nodes with the minimum ID of each link are compared. The link with the smallest value for the minimum ID node of the link is eliminated. If the minimum is defined by a common node of the both links, the elimination is based on maximum ID node.

Let us consider

$f(u,v) = (b(u,v), \min(id(u), id(v)), \max(id(u), id(v)))$, and the order relation $\leq$ defined on triples:

$$(x,y,z) \leq (x',y',z') \Leftrightarrow \quad x < x' \vee (x = x' and y < y') \vee (x = x' \wedge y = y' \wedge z < z'). \quad (1)$$

By applying all the three criteria, we are assured that all the triangles are eliminated, and none of the 1-hop neighbors is also in the 2-hop neighbors list.

Similar with the properties of a RNG graph, G' preserves the connectivity and the maximum bandwidth paths between any two vertices, while reducing the density of the graph.

The heuristic is presented in *Algorithm 1*.

Algorithm 1 Graph density reduction

Let $N(u) = \{n_1, n_2, \ldots, n_n\}$ be the list of 1-hop neighbors of the current node u.

```
function GET_BWRNG(u)
    N'(u)=N(u)
    for each v in N' do
        for each w in N(v) ∩ N(u) do
            if f(u,v) < f(u,w) ∧ f(u,v) < f(w,v) then
                remove v from N'(u)
                break
            end if
        end for
    end for
return N'(u)
end function
```

B. Advertised neighbor set selection

From the reduced graph, we will select the neighbor set that preserve maximum bandwidth paths. It is computed by each node, base on 2-hop neighbors information.

The 1-hop neighbors are evaluated in the descendant order of the bandwidth of the link with the current node, u. A 1-hop neighbor of u, n_i is added to the set of advertised neighbors A only if it provides a maximal bandwidth path between the node u and at least one of its 2 hop neighbors. The evaluation stops when all the maximal bandwidth paths between the node u and the 2 hop neighbors are found.

Let n_j be the 1-hop neighbor that represents the path with maximum bandwidth between u and the 2 hop neighbor n'_i. It is equivalent with:

$$\min\{b(u,n_j), b(n_j,n'_i)\} \geq \min\{b(u,n_k), b(n_k,n'_i)\}, \quad \forall k = \overline{1:n} \wedge n_k \in N(u) \cap N(n'_i)$$

This relation is used to evaluate each 1-hop neighbor. Algorithm 2 returns the set of neighbors defining maximum bandwidth paths.

Algorithm 2 Select advertised neighbors set

Let $N(u) = \{n_1, n_2, \ldots, n_n\}$ be the list of 1 hop neighbors of u.

```
procedure GET_BW_QANS(u)
    Start with empty sets A and N'_j.
    for each 2 hop neighbor n'_i do
        determine b_max(u, n'_i)
    end for
    for each node n_j ∈ N(u) do
        for each node n'_i in N(N(u)) ∩ N(n_j) do
            if b(u, n_j) ≥ b_max(u, n'_i) then
                if b(n_j, n'_i) ≥ b_max(u, n'_i) then
                    add n'_i to N'_j
                end if
            end if
        end for
        if N_j not empty then
            add n_j to A.
        end if
    end for
end procedure
```

There can be more than one maximum bandwidth path to a 2 hop neighbor in the selected set A. Each 1-hop neighbor n_i will define a maximum bandwidth path for a set N'_i of neighbors such that:

$$\bigcap_{i=1}^{n} N_i = N(N(u)).$$

In order to further optimize the dimension of QANS sets, we consider the following greedy method (implemented by algorithm 3), for removing nodes providing redundant paths. At the beginning both the set A'of neighbors and the set N' of 2-hop neighbors covered by the nodes in A' are empty. Each time the node from A that provides the greatest number of maximum paths to 2 hop neighbors not already in N' is added to A' and the covered neighbors in N'. The selection stops when all the 2 hop neighbors are covered. A' will represent the QANS set. An example of selection for the presented algorithm is shown in Fig. 1. After the evaluation of all links bandwidth of the graph in 1(b), only n_2 and n_4 are selected in 1(c).

Algorithm 3 Optimized advertised neighbors set

Start with empty sets A' and N'.
procedure REDUCE_BW_QANS(u)
  while $N' \neq N$ **do**
    Add to A' n_j for which

$$N_j/N' = \max_{0 \leq i < n} N_i/N'$$

.
    Add elements from N_j to N'.
  end while
end procedure

C. Proof of correctness

We have to prove that our algorithm 3 generates topology information which are sufficient to compute maximum bandwidth paths. We can notice that this statement is only needed for nodes which are not directly connected. In order to obtain this proof of correctness, we use three steps: (a) prove that the *graph density reduction* preserves maximum bandwidth (this property includes connectivity preservation), (b) prove that *advertised neighbor set selection* preserves maximum bandwidth between 2-hop neighbors, and (c) prove that 2-hop maximum bandwidth preservation is enough to guarantee maximum bandwidth preservation for any couple of nodes distant of at least two hops.

Concerning *graph density reduction*, we show that for all couple of nodes (u, v) and paths p between u and v in G, then there exist a path p' between u and v such that $b(p) \leq b(p')$. For a path $p = \{a_0, a_1, \ldots, a_k\}$ in G, we show how to build the path p'. Let us consider removed edges in ascendant order (according to the order defined in eq. 1). Each time that an edge (x, y) contained in p is removed, we apply the following operation. If (x, y) is deleted from the initial graph, it means that there exist two links (x, z) and (z, y) such that $f(x, y) < f(x, z)$ and $f(x, y) < f(z, y)$. By definition of the function f and of the order, it implies that $b(x, z) \geq b(x, y)$ and $b(z, y) \geq b(x, y)$. Moreover, these two links have not been removed yet and we can simply replace the sub-path $\{x, y\}$ by $\{x, z, y\}$. Since the number of edges is finite, when the process ends, we have a path with higher or equal bandwidth.

For the optimality of our *advertised neighbor set selection* algorithm for 2-hops neighbors in G', it suffices to observe that maximum bandwidth paths in G' between 2-hops neighbors cannot be longer than two hops. Let us consider a loop-free path $p = \{a_0, a_1, \ldots, a_k\}$ in G between $u = a_0$ and $v = a_k$, one of its 2-hops neighbors in G, such that $\forall 1 \leq i < k$ the intermediate node a_i in a 1-hop neighbor of u in G. We show that k is equal to two. Indeed, if k is greater than 2, it means that a_2 is a 1-hop neighbor of u. It implies that the edges (a_0, a_1), (a_1, a_2) and (a_0, a_2) exist in G. However, triangles cannot exist in G because at least one of the edges satisfies the condition to be removed compared to the two other ones. Because our algorithm preserves maximum bandwidth 2-hop paths, it is enough to guarantee bandwidth preservation between 2-hop neighbors.

Now, we show that the knowledge of maximum bandwidth path between 2-hop neighbors is enough to compute maximum bandwidth path between two arbitrary nodes distant of at least two hops. More precisely, for a loop-free path $p = \{a_0, a_1, \ldots, a_k\}$ in G with $k \geq 2$, we show by induction that that we can compute a path p based on 2-hop maximum bandwidth path such that $b(p) \leq b(p)$. If $k = 2$, the property simply holds because of previous statement. If $k > 2$, we know by induction that the subpath $p_1 = \{a_0, \ldots, a_{k-1}\}$ can be replaced by a subpath $p_1 = \{b_0, \ldots, b_l\}$ which use only knowledge of 2-hop maximum bandwidth path and such that $b(p_1) \leq b(p_1)$ (note that we have $a_0 = b_0$ and $b_l = a_{k-1}$). Because G does not contains triangles, the node b_{l-1} in p_1 is a 2-hop neighbor of a_k. From induction hypothesis, the subpath $\{b_{l-1}, b_l = a_{k-1}, a_k\}$ can be replaced by a 2-hop maximum bandwidth path $\{b_{l-1}, c, a_k\}$. In conclusion, we can compute a path $p = \{a_0 = b_0, b_1, \ldots, b_{l-1}, c, a_k\}$ with a higher of equal bandwidth.

These steps are enough to show that our algorithm guarantees bandwidth optimality for nodes distant of at least 2-hops (in G or G since G is a reduced graph of G). The proof of this optimality is simplified because of the use of G which does not contains triangles.

V. Topology filtering for delay

A. Graph density reduction

Delay is another demanding constraint for QoS routing, especially in the case of multimedia applications. The difference is that the delay of each link is added to the overall value.

For evaluating delay constrained routing we will use the same representation of a network by the graph $G = (V, E)$. If D is the value of the maximum delay link, then a link's delay value is defined by a function d defined on the set of edges E with values in the interval $[0, D]$. The delay is an additive metric. This means that for a path p between nodes u and v,

$$p = \{u, u_1, u_2, \ldots, v\},$$

the delay d_p is defined on $[0, D_p]$ and is

$$d_p = d(u, u_1) + d(u_1, u_2) + \ldots + d(u_n, v).$$

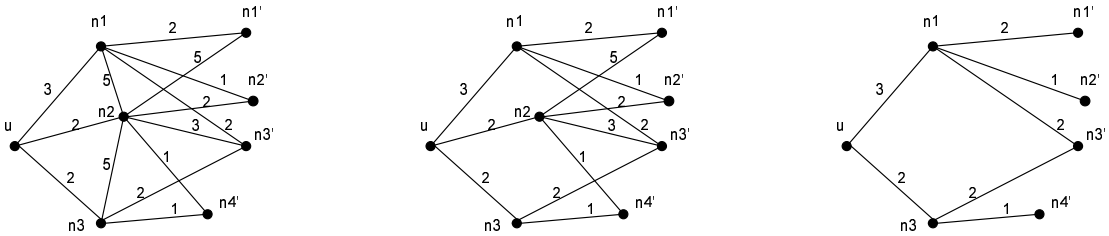


Fig. 2. Example of delay QANS selection for a node

For reducing the density of the graph we consider again the case of a triangle in the network, generated by u, a common neighbor of n_1 and of n_2, also neighbors.

Let $u, n_1 \, and \, n_2 \in V$ such that $(u, n_1), (n_1, n_2)$ and $(n_2, u) \in E$. Similar with the bandwidth we will reduce the density of the graph by removing the worst performance edge from the triangles generated by 1 hop neighbors. An edge is the worst performance edge if it has a delay greater or equal than a 2 hop path between the same nodes. An worst performance edge (u, n_1) is characterized by the property: $\exists n_2 \in V$ such that $d(u, n_1) \geq d(n_1, n_2) + d(u, n_2) \, if \, d(n_1, n_2) \neq 0 \, and \, d(u, n_2) \neq 0$.

Algorithm 4 Graph density reduction

Let $N(u) = [n_1, n_2, \ldots, n_n]$ be the list of 1 hop neighbors of u.
Let N'_j be the set of 2 hop neighbors covered by n_j.

```
function GET_DELAYREDUCEDGRAPH(u)
    N'(u) = N(u)
    for each v in N'_u do
        for each w ∈ N(v) ∩ N(u) do
            if f(u,v) ≥ f(u,w) + f(w,v) then
                remove v of N'(u)
                break
            end if
        end for
    end for
return N'(u)
end function
```

By removing all the edges (u, n_1) with the property above from E, nor the connectivity neither the values of minimum delay paths are not affected.

Similar with the RNG, removing the greatest delay edge from a triangle does not influence the connectivity of the graph. If one of the edges has a delay equal with 0, then the other two links will be both removed. This situation is avoided by imposing the last condition.

In order to discuss the preservation of minimum delay paths value, we will consider a graph, G' obtained by removing all the edges in E with the property above. If the set of minimum delay paths is represented by P, then $\forall p \in P$, $\exists p'$ in P', the set of minimum delay paths in G' such that $d_p(p') = d_p(p)$. Indeed, if $d(n_i, n_{i+1}) \geq d(n_i, n'_i) + d(n'_i, n_{i+1})$, for each path $p = \{u, n_1, n_2, \ldots, n_i, ni + 1, \ldots, v\} \, in \, P$, there is a path $p' = \{u, n_1, n_2, \ldots, n_i, n'_i, n_{i+1}, \ldots, v\} \, in \, P$ with the property that $d_p \geq d_{p'}$.

B. Advertised neighbor set selection

The next step is to select the subset QANS of nodes of G' that provides complete network connectivity through minimum delay links. Although the procedure above will not remove all the triangles from the network, it assures us that when they still exists, the minimum delay path is the direct one. Therefore, in order to find the QANS set, is necessary to remove from the list of 2-hop neighbors of u, those that are also 1-hop neighbors.

Similarly with the first algorithm, a 1-hop neighbor of u, n_i is added to the set A only if it provides a minimum delay path between the node and at least one of its 2 hop neighbors. The algorithm stops when all 1-hop neighbors are evaluated.

Algorithm 5 Select advertised neighbors set

Let $N(u) = [n_1, n_2, \ldots, n_n]$ be the list of 1 hop neighbors in G'.
Let N'_j the set of 2 hop neighbors covered by n_j: $N'_j = N(N(u)) \cap N(n_j)$

```
procedure GET_DELAY_QANS
    start with empty sets QANS and N'_j.
    for each 2 hop neighbor n'_i do
        determine d_min(u, n'_i)
    end for
    for each node n_j ∈ N(u) do
        for each node n'_i in N(n_j) do
            if d(n_j, n'_i) + d(u, n_j) = d_min(u, n'_i) then
                add n_j to N'_j
            end if
        end for
        if N_j not empty then
            add n_j to QANS.
        end if
    end for
end procedure
```

The selected set will preserve the minimum delay paths. For each path p in the graph G, we can build a path p' in the graph G', with the length smaller or equal to the length of p and with the same delay.

Let $p = \{u, n_1, n_2, \ldots, n_{i-1}, n_i, n_{i+1}, \ldots, v\}$. Let us suppose that a node n_i it is not in QANS subset of n_{i-1}. Then it exists n'_i such that $n'_i \in$QANS and the delay $d_p((n_{i-1}, n'_i), (n'_i, n_{i+1})) \leq d_p = ((n_{i-1}, n_i), (n_i, n_{i+1}))$.

There can be more than one minimum delay path to a 2 hop neighbor in the selected set QANS. This means that the QANS set can be further minimized. We consider the same greedy method for selecting a smaller set. At each step the 1-hop neighbor that covers the maximum number of 2 hop neighbors not covered yet is selected. The selection stops when all the 2 hop neighbors are covered. The algorithm is identical with the bandwidth case.

Fig. 2 illustrates an example. The initial graph is represented in 2(a). In 2(b) the links with the worse performance metric are eliminated. In 2(c) is selected the minimum set of neighbors on best performance paths to the 2-hop neighbors set.

VI. Simulation

We implemented a simulator to evaluate the performances of the proposed algorithm. Tests were made with a static network of 200 nodes. Nodes are randomly distributed in order to obtain a given average number of neighbors. We compare our algorithm to QOLSR protocol.

Both QOLSR and OLSR-QANS are enhancements to OLSR protocol and aim at providing QoS routes. In a proactive protocol, each node declares the links with its neighbors, by sending broadcasts into the network. Network traffic is influenced by the size of packets and the number of broadcasts. The size of packets depends on the number of declared links. The number of broadcasts depends on the number of neighbors selected by a node to retransmit a message. We will compare the subset of neighbors selected for QoS routing and for network control messages retransmission. QoS performances are evaluated by the number of paths, that respect the QoS requirements, successfully found.

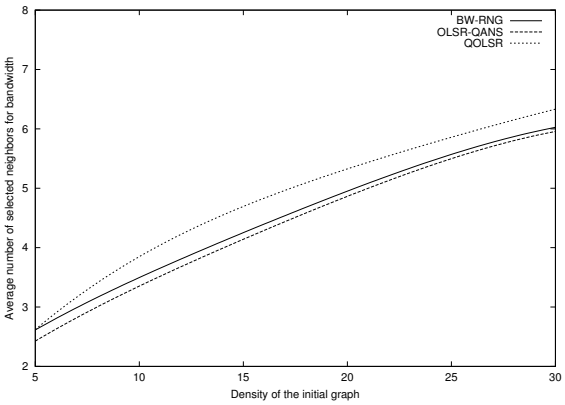


Fig. 3. Maximum bandwidth neighbors selection

We computed the number of neighbors selected to route messages. Fig. 3 compares the average number of 1-hop neighbors used for QoS path. The metric used is the bandwidth. The average size of 1-hop neighbors in the bandwidth RNG graph is smaller than the QOLSR selection. Accordingly, the 1-hop set selected by OLSR-QANS is smaller than QOLSR selection for bandwidth with 12%.

Fig. 4 compares the number of nodes selected for broadcasting network information. Our protocol uses MPR sets for broadcasting, while QOLSR uses the same set of nodes as the one for QoS paths. MPR sets are smaller than QOLSR because they have only the constraint of 2-hop neighbors to cover. QOLSR selection has to fulfill additional requirements imposed by the QoS metric.

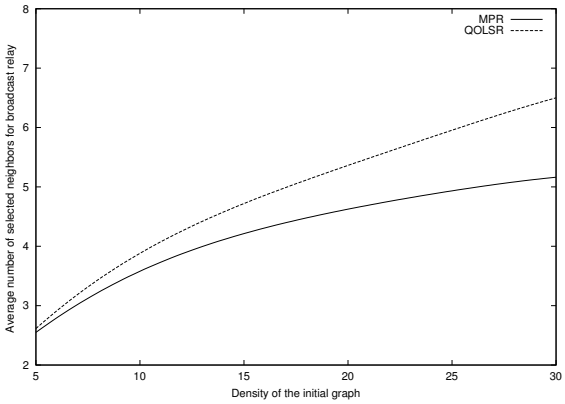


Fig. 4. Broadcast forwarding neighbors selection

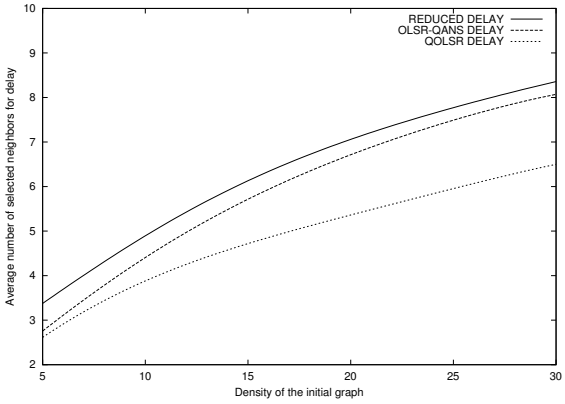


Fig. 5. Minimum delay neighbors selection

In Fig. 5 are presented the results of selection for delay. The selection of QOLSR is smaller with 18%. The size of 1-hop set in the reduced graph for delay is influenced by the conditions imposed to worse performance links, which are more restrictive than in the case of bandwidth.

In Fig. 6 we analyse the performances from the point of view of the bandwidth metric requirements. We present the dependence of path bandwidth on the average density. Paths are computed with a Dijkstra algorithm modified for concave constraints. The bandwidth gain obtained by using QoS protocols in OLSR-QANS compared with the bandwidth of the path in the QOLSR graph is relatively constant and has the average value of 8%. The bandwidth gain is obtained with a smaller set of 1-hop neighbors.

Similarly, fig. 7 shows the raport between the delay obtained for paths computed in the case of the two protocols. Paths are computed with Dijkstra algorithm, that considers the delay as the cost associated to links. The raport between the delays depends on the density of the network. For densities greater than 20, minimum delay of the paths in OLSR-QANS graph

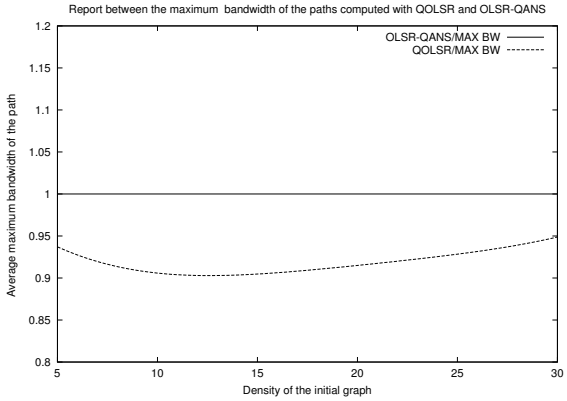


Fig. 6. Path average bandwidth comparison

is with 30% smaller than in QOLSR graph. This is obtained with the increase of 18% in the number of 1-hop neighbors used for QoS routing.

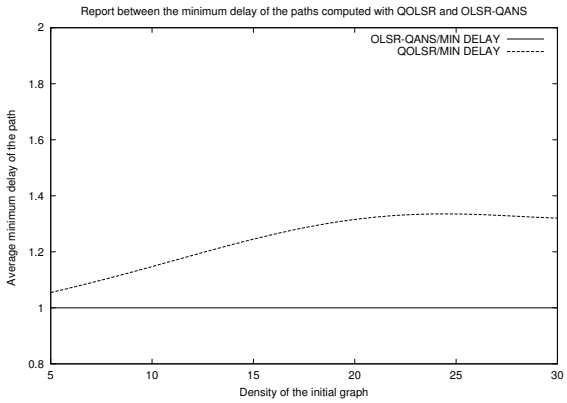


Fig. 7. Path average delay comparison

A concern in QoS routing is route computation. The length of the paths is influenced by the elimination of both links and nodes from the initial graph. We compared the distorsion of maximum bandwidth paths for the two protocols. For bandwidth the routes computed with QOLSR are smaller, as it can be seen in Fig. 8. For delay, the distorsion is influenced by the density of the graph, for higher densities, the distorsion of OLSR-QANS becomes smaller than QOLSR, as can be seen in Fig. 9 .

VII. Conclusions

In this paper we presented a QoS routing protocol. It is an extension of OLSR, a proactive routing protocol for MANET. We presented the modifications made to packets structure and the set of nodes selected for forwarding the messages. We explained the algorithm used to select the set of neighbors that respects the QoS requirements and we proved the correctness of the selection methods. Then we compared it with another extension of OLSR for QoS routing, QOLSR. The results shows that we obtained better performances in terms of QoS metric than QOLSR and a smaller number of broadcasts. Like all the other QoS protocols, our protocol has the drawback of

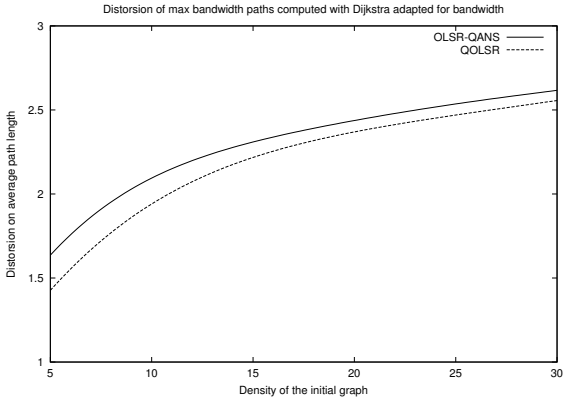


Fig. 8. Distorsion of the length of the maximum bandwidth paths

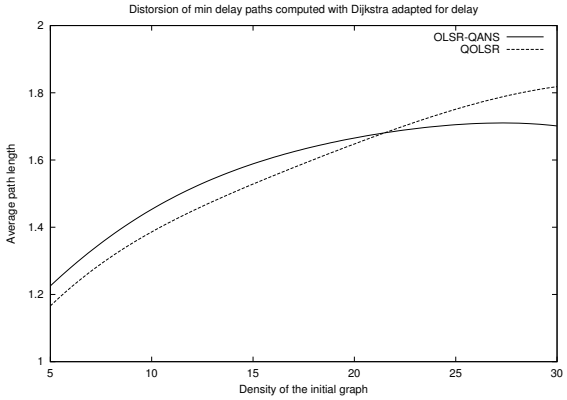


Fig. 9. Distorsion of the length of the minimum delay paths

routing QoS compliant packets on paths with a greater length that the best effort ones. Future works include the evaluation of the protocol when both bandwidth and delay are considered.

References

[1] "Mobile ad-hoc network ietf working group." [Online]. Available: http://www.ietf.org/html.charters/manet-charter.html

[2] B. R. E. Crawley, R. Nair and H. Sandick, "A Framework for QoS-based Routing in the Internet," RFC 2386, Aug. 1998. [Online]. Available: http://rfc.net/rfc2386.html

[3] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626 (Experimental), Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[4] L. V. Amir Qayyum and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobilewireless networks," in *Proceedings of the 35th Hawaii International Conference on System Sciences*, vol. 09.

[5] I. Jawhar and J. Wu, *Quality of Service Routing in Mobile Ad Hoc Networks*, 2004.

[6] I. Gerasimov and R. Simon, "A bandwidth-reservation mechanism for on-demand ad hoc path finding," in *Simulation Symposium, 2002. Proceedings. 35th Annual*, Apr. 2002, pp. 27 – 34.

[7] I. Jawhar and J. Wu, "Race-free resource allocation for qos support in wireless networks," pp. 179–206, 2005.

[8] I. Gerasimov and R. Simon, "Performance analysis for ad hoc qos routing protocols," in *mobiwac*, 2002, p. 87.

[9] H. Badis and K. A. Agha, "Optimal path selection analysis in ad hoc networks," LRI: Laboratoire de Recherche en Informatique, Tech. Rep., Aug. 2004.

[10] P. Sinha, R. Sivakumar, and V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm," in *INFOCOM (1)*, 1999, pp. 202–209. [Online]. Available: citeseer.ist.psu.edu/article/sinha99cedar.html

[11] G. Toussaint, *The relative neighborhood graph of a finite planar set*, 1980, pp. 261–268.

AdTorrent: Delivering Location Cognizant Advertisements to Car Networks

Alok Nandan, Saurabh Tewari, Shirshanka Das, Mario Gerla, Leonard Kleinrock

Computer Science Department
University of California Los Angeles
Los Angeles, CA 90095-1596
{alok, stewari, shanky, gerla,lk}@cs.ucla.edu

***Abstract*— AdTorrent is an integrated system for search, ranking and content delivery in car networks. AdTorrent builds on the notion of Digital Billboards, a scalable "push" model architecture for ad content delivery. We present a detailed analysis of the performance impact of key design parameters such as scope of the query flooding on the query hit ratio. Our mobility model for the urban, vehicular scenario can be used in conjunction with the analytical model for estimating query hit ratio by a system designer to determine the scope of the query flooding as a function of the available storage per vehicle for their application.**

I. Introduction

One of the most important sources of revenue for big Internet-based companies are advertisements. With vehicular networks poised to become part of the Internet, this new "edge" of the Internet represents the next frontier that advertising companies will be striving to reach. As advertisers struggle to reach increasingly distracted and jaded American consumers, they have sought nontraditional media for their advertisements (Ads), from elevators to cell phone screens.

Content-targeted advertising paradigm has proved to be a resounding success in advertising on the conventional Internet. As the Internet expands to mobile devices, even vehicular nodes are becoming a part of the "edge" of the Internet. Several interesting challenges in application design arise, while designing a targeted ad delivery mechanism for cars.

Consider this scenario: you are driving on Interstate-5 from Los Angeles to San Francisco to visit relatives. On the way, you realize that you need to buy some gift for them. You initiate a search for "new DVD releases". The Ad software is not only keyword aware but also *location* aware. Hence the search results return not only the content or latest DVD releases but also the latest deals on those DVD releases in stores.

Imagine another similar scenario: you are traveling to Las Vegas and are 50 miles from the city. You want to search for all hotels in the vicinity that cost less than 200 dollars per night, preferably with virtual tours of the hotels.

AdTorrent seeks to provide to the user, relevant Ads guided by a particular keyword search. Ads potentially can be multimedia clips, for example, virtual tours of hotel rooms, trailers of movies in nearby theatres or a conventional television ad.

A. Our Contributions

Vehicular Ad Hoc Networks (VANETs) present interesting challenges to protocol design. One of the key differentiating characteristics is the time-varying nature of vehicle densities and the mobility model. Mobility has an important impact on application design.

The contributions of this paper are as follows: (1) firstly, we propose a novel push-model based location-aware ad service architecture, designed for vehicular environments (2) secondly, we present a group mobility model for urban vehicular traffic, (3) thirdly, we present a peer-peer protocol that enables efficient keyword-set based search and quickly delivers top ranked content to the end user using swarming, (4) finally, we present a model for hop limit selection of search query flooding in the AdTorrent network. Our results on the optimal hit rate and the cache probability distribution that maximizes the overall hit rate as a function of the hop limit of query flooding is applicable in any hop-limited query flooding application. Our analysis of hit rates for LRU-based cache management extends previous work in the area by including the effect of swarming on the steady-state cache probabilities.

B. Organization of the Paper

The rest of the paper is organized as follows. Section II describes the Vehicular communication architecture. Section III gives an overview of the operation of the ad service in a vehicular scenario. Section IV describes the novel mobility model we used for the purpose of evaluation and details our evaluation of the performance of our protocol using simulation. Section V gives a brief overview of AdTorrent, a push-model of content dissemination based on a popular swarming protocol. Section VI describes the model for hop limit selection where we derive the maximum hit rate achievable for a specific hop limit as a function of the cache size and describe our model for computing hit rates in a swarming-based content delivery scenario when the underlying cache management is based on LRU. We outline the related work in section VII. Finally, Section VIII concludes the paper.

II. Preliminaries

In this section we describe the vehicular environment and the assumptions about the environment we used to design our

protocol.

The network consists of a set of N nodes with same computation and transmission capabilities, communicating through bidirectional wireless links between each other, this is the infrastructure-less ad-hoc mode of operation. There are wireless gateways at regular intervals providing access to the rest of the Internet using infrastructure support (either wired or multi-hop wireless). We assume a CSMA/CA MAC layer protocol (IEEE 802.11a) that provides RTS/CTS-Data/ACK handshake sequence for each transmission.

Our vehicular wireless architecture is composed of two kinds of communications, namely, vehicle-vehicle and vehicle-gateway. Dedicated Short-Range Communication (DSRC) [4] is a short to medium range communication technology operating in the 5.9 GHz range can be used for vehicle-vehicle communication. For a more detailed description of the DSRC characteristics, we refer the reader to [8].

- *Data is not strictly real-time:* There are no real-time constraints on the data, thus in some sense, the data is delay-tolerant.
- *Data is meta-tagged:* Meta-data can be the file-name, the format and/or key-words extracted from the data. For some types of data, such as text documents, metadata can be extracted manually or algorithmically. Some types of files have metadata built-in; for example, ID3 tags on MP3 music files.
- *Communication between vehicles is over a low data rate connection:* This constraint depends on the radio technology used. Currently, 802.11x devices will offer *goodput* of the order of a few hundred Kbps.
- *Push model:* Data is being continually "pushed" by the access points to the nodes in the transmission range.
- *Multi-hop delivery:* It is infeasible to transmit data to more than a few hops.

III. The Digital Billboard Architecture

The digital billboard architecture serves to deliver Ads to the vehicles that pass within the range of the Access Points (APs). This architecture is:

- *Safer:* Physical billboards can be distracting for drivers
- *Aesthetic:* The skyline is not marred by unsightly boards.
- *Efficient:* With the presence of a good application on the client (vehicle) side, users will see the Ad only if they actively search for it or are interested in it.
- *Localized:* The physical wireless medium automatically induces locality characteristics into the advertisements.

Every Access Point (AP) disseminates certain sets of Ads that are relevant to the proximity of the AP deployment. This is reasonable since it is the extension of the physical billboards that we very often see lined on the streets and freeways that advertise the best offers available at the next restaurant. Our AP acts as a "digital billboard". This model makes sense economically as well since business owners in the vicinity subscribe to this digital billboard service for a fee. The APs continually disseminate these advertisements to the vehicles that traverse the coverage area. The dissemination rate can be determined by different *levels of service* demanded and paid for by the billboard owner.

Leveraging this architecture, we want to design a location-aware distributed mechanism to search, rank and deliver content to the end-user (the vehicle). We focus not only on simple text-based Ads but also on larger multimedia Ads, for example, trailers of movies playing at the nearby theater, virtual tours of hotels in a 5 mile radius, or conventional television advertisements relevant to local businesses.

Every node that runs the application collects these advertisements and indexes the data based on certain meta-data which could be keywords, location and other information associated with the data. We assume that Ads are uniquely identifiable using a document identifier (DocId).

In the next section we describe a group mobility model for an urban vehicular network. The model guides us in the design of AdTorrent, a protocol for advertisement search and delivery on the vehicular network.

IV. Mobility Model

The mobility model is designed to describe the movement pattern of mobile users, and how their location and velocity change over time. Mobility patterns play a significant role in determining the protocol performance and thus are an important parameter to the protocol design phase. It is important for mobility models to emulate the movement pattern of targeted real life applications in a reasonable way. Otherwise, the observations made and the conclusions drawn from the simulation studies may be misleading. Thus, when evaluating our vehicular ad hoc network protocol, it is imperative to choose the proper underlying mobility model. Different application scenarios lend themselves to different mobility models. For example, a campus-wide wireless network deployment will see different mobility patterns (less constrained, more random) than an urban vehicular grid scenario (low entropy of vehicles, group mobility).

In modeling and analyzing the mobility models in a VANET, we are more interested in the movement of individual nodes at the microscopic-level, including node location and velocity relative to other nodes, because these factors directly determine when the links are formed and broken, since the communication pattern is peer-to-peer.

We used the US Census bureau data for street level maps. As a starting point, using the methodology from [11], we generate the mercator projection of the data, in our case the local map of an area around UCLA in Fig. 1.

However, in [11], the actual mobility model is quite similar to the Random Waypoint model in the sense that the vehicle's arrival and direction and speed are similar to the Random Waypoint model. This results in the vehicular mobility model being very similar to the Random Waypoint model. In reality, a complex mobility behavior is observed. Some nodes move in groups; while others move individually and independently; a fraction of nodes are static. Moreover, the group affiliation

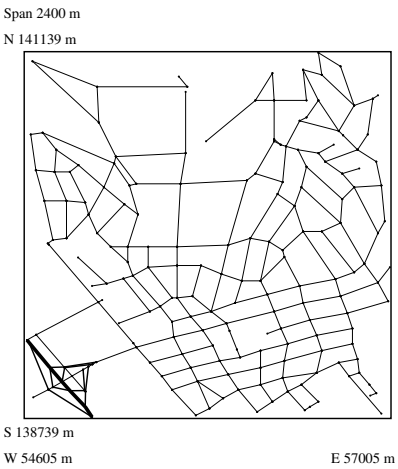


Fig. 1. Local Map of Westwood, an area around UCLA

is not permanent. The mobile groups can dynamically re-configure themselves triggering group split and mergence. All these different mobility behaviors coexist in vehicle or urban scenarios. We refer to the non-uniform, dynamic changing scenario described above as "heterogeneous" group mobility scenario [15]. A good realistic mobility model must capture all these mobility dynamics in order to yield realistic performance evaluation results, which, unfortunately, is not satisfactorily captured in any of the existing models.

We propose a "real track" based group mobility model (RT model) that closely approximates the above "heterogeneous" mobility patterns happening in the scenarios of vehicle ad hoc networks. It models various types of node mobility such as group moving nodes, individually moving nodes as well as static nodes. Moreover, the RT model not only models the group mobility, it also models the dynamics of group mobility such as group merge and split.

The key idea of our proposed model is to use some "real tracks" to model the dynamics of group mobility. In our simulation scenarios, these "real tracks" are derived from the streets from actual maps. The grouped nodes must move following the constraint of the tracks. At the switch stations, which are the intersections of tracks/streets, a group can then be split into multiple smaller groups; some groups may be even merged into a bigger group. Such group dynamics happen randomly under the control of configured split and merge probabilities.

Nodes in the same group move along the same track. They also share the same group movement towards the next switch station. In addition, each group member will also have an internal random mobility within the scope of a group. The mobility speeds of these groups are randomly selected between the configured minimum and maximum mobility speeds. One can also define multiple classes of mobile nodes, such as pedestrians, and cars, etc. Each class of nodes has different requirements: such as moving speed etc. In such cases, only nodes belonging to the same class can merge into a group.

Groups split and merge happen at the switch stations. Each group is defined with a group stability threshold value. When at the switch stations, each node in the group will check whether its stability value is beyond its group stability threshold value. If it is true, this node will choose a different track from its group. A group split happens. When several groups arrive at the same station and select the same track for the next movement, naturally, they will be merged into one bigger group.

The proposed RT model is also capable of modeling randomly and individually moving nodes as well as static nodes(such as sensors). Such non-grouped nodes are not restricted by the switch stations and real tracks. Instead, their movements are modeled as random moves in the whole field.

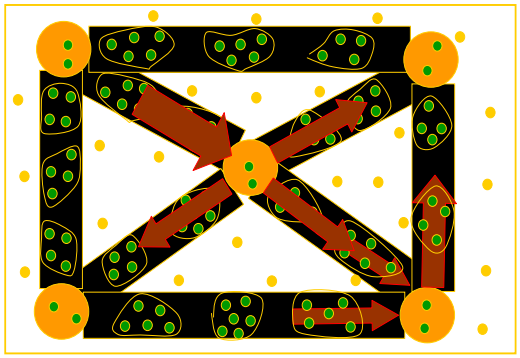


Fig. 2. Overview of Real Track Based Group Mobility Model.

Fig. 2 illustrates a main idea of the proposed real track based group mobility model. In this example, group moving nodes are moving towards switch stations along the tracks. They split and merge at switch stations as shown in the figure. The black nodes in Fig. 2 represent the individually moving nodes and static nodes. They are placed and move independent of tracks and switch stations.

We evaluate the scenarios along the following metrics as defined in [11]. For brevity we present only the average connectivity duration metric, which is the most essential for protocol design in our scenario.

Average connectivity duration: This is the duration of the time two nodes have a path between them. We further quantify this metric based on the maximum allowable hops for any path between the two nodes. This metric is relevant to our application as it justifies the usage of a swarming content delivery model in the presence of limited connectivity between the nodes.

We used a 500m transmission range for the radios. In our case we adjusted the number of nodes, to 30, 50 and 60, spread over an area of $2400m \times 2400m$. The average number of nodes in the transmission range were 4.1, 6.9 and 8.1 respectively. Each run of simulation were 900s long. Also we evaluate the scenarios at regular intervals of 10s.

We observe from Fig. 3 that for a 4-hop limit path the connectivity duration has an almost 100% increase as opposed

to a 3-hop limited path. Longer connectivity durations lead to robust protocol performance (since the initiated downloads have a higher chance of being completed). For urban vehicular scenarios, the results in Fig. 3 suggest that the incremental gain from increasing the hop limit up to 4 might be useful for increasing the robustness of protocol performance.

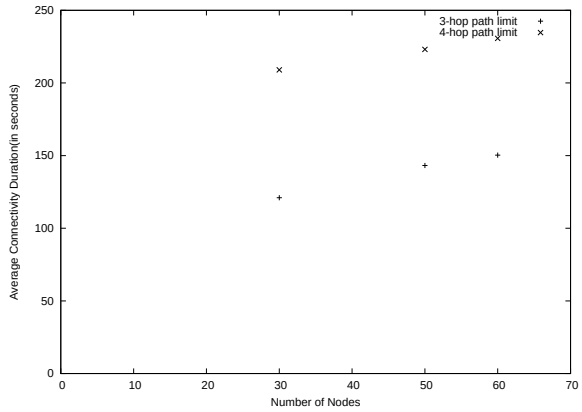


Fig. 3. Average Connectivity duration

V. AdTorrent: Design

We outline the primary design goals of *AdTorrent* as follows:

1) location aware torrent[1] ranking algorithm;
2) search should be simple and robust, in presence of node failures and departures;
3) leverage churn
4) minimal overhead of communication;

There are three main tasks performed by our application. Namely, *search* for relevant ad-content, *query dissemination* and *content delivery*. We address each of these functions in the following sections.

A. Search

Search involves associating keywords with document identifiers and later retrieving document identifiers that match combinations of keywords. Each file is associated with a set of metadata: the file name, its format, genre (e.g. in advertisements). For some types of data, such as text documents, metadata can be extracted manually or algorithmically. Some types of file have metadata built-in; for example, ID tags on MP3 files.

Distributed Hash Tables (DHTs) have been proposed for distributed lookups. We do not use a DHT for distributed lookup, since it is well-known that DHTs are not very stable under high churn [10]. Our query dissemination mechanism aims to achieve robustness rather than communication efficiency.

However, we introduce certain optimizations to the index information dissemination that will reduce the amount of search query communication overhead.

[1]we use the terms : document and torrent interchangeably

Indexing: Vertical partitioning divides an index across keywords. Horizontal partitioning divides an index across documents, so all the nodes have an entry for *each* keyword. Vertical partitioning minimizes the cost of searches. However, horizontal partitioned index reduces the cost of update of document. In our scenario, the number of queries will far outnumber the number of updates, since we assume the documents typically searched for, are not changing frequently. Our index is partitioned based on set of keywords. This was first introduced in KSS [5]. The motivation of using a keyword set based indexing is the reduction of overhead in terms of query data information. The downside of this approach is higher cost of insert and storage. In VANETs, we believe, storage will not be a limitation. In the AdTorrent application, one of the key characteristics will be the infrequent updates of ads, maybe once in one day or less.

The index we maintain is distributed. However, every node tries to maintain information of the all the two-hop neighborhood of itself. The documents are indexed on keys. Keys consists of SHA-1 hashes of the keywords sets. Along with the keys and the URL of the data, we also store additional meta-data associated with the data. The metadata is stored in an index corresponding to each subset of at most K metadata items. KSS uses a distributed inverted index to answer queries efficiently with minimal communication overhead. Each entry of the index contains: (1) the hash of the searchable sets of keywords as the index key, (2) a pointer to the data such as the URL of the data and, (3) meta-data associated with the data.

Placement: In a wireless scenario, it makes sense to co-locate the index and the data corresponding to the index entry. This is to reduce the overhead of data discovery latency once the index for that data has been located.

B. Query Data Dissemination Optimization

Each node disseminates the content availability information in the form of a bloom filter. Bloom filter [1] is an efficient method to test for set membership. In our case, the bloom filter is constructed to test the keyword membership for a particular node. A bloom filter is computed by each node based on the keywords related to the data, the node has stored. Since the data downloaded is only once every AP encounter or if the node explicitly downloads some swarming torrent, hence the updates of bloom filter and dissemination is not very frequent.

We now enumerate the basic steps of the algorithm.

The indexing scheme described above does not have a document ranking algorithm. The order of query results propagation and display is equally important for successful and timely dissemination in a VANET. This assumes further importance in VANET since the mobility of nodes might render some query results obsolete or irrelevant in short period of time. We incorporate a location metric in the document ranking scheme. One way to support the document ranking would be to score a document based on the following categories.

1) location
2) max # of pieces

Algorithm 1 AdTorrent: Query Processing, Ranking and Content Delivery

user_input = **search("A B C")**

num_local_entries = **lookup_local_index**(hash(AB), hash(BC), hash(CA))
if (num_local_entries $> k_1$)
    goto *LookupDone*
else
    /* Found $< k_1$ local entries */
    /* not in the 2-hop neighborhood */
    num_remote_entries = **scoped_flood**(hash(XY), m)
    /* $\forall$ XY $\in$ AB,BC,CA */
    After T_1 seconds, if NO response, return NO
    If k_1 entries are found then

LookupDone:
    /* now have k_1 entries (local or remote in 1-4 hops) */
    send_udp_ctrl(Hash(XY))→METADATA(e.g. TorrentID)

    /* Collect meta_data after T_2 */
    torrent_ranking(meta_data, params)

Step Final:
    swarm(TorrentID)
    /* returns a list of Peers & HopCounts*/
    /* (this may be beyond the the scope of the search) */
    decentralized_tracker()
    /* By allowing the list of Peers beyond the k-hop scope of the search, we add some randomization */

3) stability of neighbors
4) relevance of the DocID to the Meta-Data queried

C. Content Delivery

Once an accurate document ranking has been performed, the actual delivery of content can be done by swarming. One of the factors that determined the ranking of a document in the query results was the number of sub-pieces of the document that were available and the location of the pieces. Thus the torrent ranking guides the system to choose documents which are more amenable to swarming downloads. The vehicle now joins the existing BitTorrent-like stream to start getting pieces of the document from neighboring nodes. We propose to do this using our earlier work in [8]. Swarming allows us to be robust to node failures (cars going out of range or powering down) and efficient in terms of delivery (the cars form a sort of end-system-multicast tree). However, the success of swarming especially in wireless ad-hoc networks depends hugely upon cooperation among vehicles at both the routing layer (forwarding packets for others) and at the application layer (sharing the advertisements that have been downloaded). In the next section we address the concerns with respect to selfish behavior on the AdTorrent network and discuss ways to mitigate it.

VI. Model

As discussed in the previous section, AdTorrent searches for relevant ad-content using a hop-limited query broadcast. Since setting a large hop-limit queries more nodes, a larger hop-limit improves the probability of finding the desired content and will likely increase the number of sources from which the content may be downloaded from. However, the gains in the quality of search results comes at the cost of significant increase in the messages sent per query in the network. Since only limited bandwidth is available in wireless medium, careful analysis of this trade-off between the quality of search results and the communication costs of the search is required. We develop a simplified model of our system to explore this trade-off in this section.

Notice that if sufficient storage space is available at each node, over time as nodes make requests and download content, nearly all content will become available within a few hops. Limited storage space necessitates deletion of previously obtained content preventing accumulation of a sufficiently large corpus of the offered content. Thus, the size of the per-node local storage space is a key factor in determining the hop-limit required to achieve an overall target *hit rate* (the probability of finding the desired content).

Even when the storage space is limited, if it was possible to have a replica of all the content within a few hops from each node we could still achieve a very overall hit rate with a short hop-limit. We formalize these ideas by deriving the allocation of a given per-node storage space that maximizes the overall hit rate for a given hop limit.

While it may be possible to devise distributed mechanisms to achieve the derived optimal allocation, we note that a very high query hit rate by itself does not imply a superlative system performance. For instance, it is conceivable that after accessing a particular content, a user may wish to access a content again in near future (e.g. to compare a hotel room to a previously viewed hotel room). Typically such request patterns imply access-based cache replacement schemes such as LRU (delete the **L**east **R**ecently **U**sed file). Since, LRU is the most commonly implemented cache replacement policy, we analyze the hit rates achieved under LRU cache replacements.

The AdTorrent protocol allows a node to download from multiple sources in parallel. This parallel downloading affects access-based cache management policies like LRU as one content request by a node now has the potential of affecting the LRU ranking of the requested content at many nodes (up to the maximum number of parallel download sessions allowed in the protocol). Therefore, our analysis must evaluate the effect of downloading from multiple sources on the overall hit rates.

We assume that there are N unique files in the system (the term file represents any ad that would be downloaded), each with an associated request rate λ_i for file i per node (the

request rates are *uniform* across nodes[2]). We assume that each file is of equal size[3]. Nodes have finite local storage space to store content files. We assume that the storage space at each node is equal[4] and has the capacity to store B files. Throughout our analysis in this section, we assume that a query is for a particular file[5]. We assume that a query flood of hop limit k reaches $M(k)$ peers.

We use the following notation for the system parameters in our model:

- k = Number of hops in the search query dissemination
- M = Number of nodes in the search range
- N = Number of unique files in the system
- B = Per-node storage size in number of files
- i = File Id
- λ_i = Request rate of file i per node
- $\lambda = \sum_{i=1}^{N} \lambda_i$
- S = Swarming parameter (the maximum number of peers a peer can download from in parallel)
- j = Location in the local cache

Content list aggregation: Note that the aggregation of the content list of neighboring peers is not explicitly included in our model as the relevant hit rates can be trivially obtained from our model that assumes no aggregation. The aggregation of the content list of 1-hop neighbors implies that, with a flood of hop-limit k, we obtain the information about nodes within the hop distance $k+1$. In other words, if we find a hop-limit of $k+1$ to provide acceptable hit rates in the model described herein, AdTorrent will have the same performance with a hop-limit of k. We would like to add that while content list aggregation is a good way to improve the search performance, we believe that aggregating content lists beyond 1-hop neighbors is unwise. In a mobile wireless scenario, neighbors can change frequently and, while updating for the content list of 1-hop neighbors in case of a change is easy, keeping the content list accurate for neighbors more than 1-hop away will require costly change propagation.

For a VANET scenario, our real-track mobility model [15] is an ideal choice. The model can be run with the expected user density and an empirical expression of $M(k)$ can be obtained from the collected statistics. In our investigations, we use two different scenarios: an *dense urban scenario* and a sparser *highway scenario*. In the dense urban scenario, the growth model is grid-like and we obtain $M(k) = \alpha k^2$ with $\alpha = 4$ (our mobility model simulations for the 30 nodes in 2400m x 2400m area case in Section IV gave $\alpha = 4$). For the sparse highway scenario, we obtained $M(k) = \alpha k^{1.4}$ with $\alpha = 4$. In Table 1, we show the growth in number of nodes queried (which is directly proportional to the communication cost).

TABLE I

GROWTH RATE IN DENSE-URBAN AND SPARSE-HIGHWAY MODELS

Hop Limit	1	2	3	4	5	6	7
$M(k) \approx 4k^{1.4}$	4	11	19	28	38	49	61
$M(k) \approx 4k^2$	4	16	36	64	100	144	196

In most web and multimedia applications, different objects have been found to have very different popularity and we expect the same in our Ad-content distribution scenario. Skewed file popularity distribution is typically modeled by a Zipf-distribution and we will use the same model in our investigations.

A. Hit Rate Optimization

When the query flood has a hop-limit of k, the hit rate for file i, $H_i(k)$, is

$$H_i = 1 - (1 - p_i)^{M(k)}$$

The overall hit rate H can be written as

$$H = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} H_i = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} [1 - (1 - p_i)^{M(k)}]$$

$$H = 1 - \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} (1 - p_i)^{M(k)}$$

As the hit rates can always be increased if more storage space was available, our optimization is under the the constraint of the available storage space. Since each peer is identical in our model, the file replicas of a file will be uniformly distributed in the network and, for our purposes, it is sufficient to model the probabilities of each file being in the cache. Therefore, for our optimization, we can write the following constraint

$$\sum_{i=1}^{N} p_i = B$$

The Lagrangian for our problem is

$$G = 1 - \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} (1 - p_i)^{M(k)} + \gamma \big(\sum_{i=1}^{N} p_i - B \big)$$

where H is given in the equation above. Optimizing the hit rate w.r.t p_i gives the optimal p_i to be

$$p_i = 1 - \frac{(N - B) \lambda_i^{-\frac{1}{M(k)-1}}}{\sum_{i=1}^{N} \lambda_i^{-\frac{1}{M(k)-1}}} \ \forall \ i$$

[2]We note that the search is localized over a small geographic area so node interests are not necessarily very different. We believe that the uniform request rates assumption provides an adequate average case analysis (i.e. a more accurate model allowing for variations in request rates of files across nodes where the average per-node request rate is λ_i would not give results that are qualitatively any different).

[3]While file sizes can be different, cache replacement is implemented for fixed size data blocks. Thus, any inaccuracy in the analytical model is on account of correlated requests for the disk blocks and not the equal file size assumption. We do not believe correlated requests make a qualitative difference in the results of the model.

[4]The storage capacity in question is the capacity allocated for the push-model data storage and we expect that very few users will allocate more storage than the minimum required by the AdTorrent application.

[5]Even though a query has a set of keywords, a typical user is looking for a particular item when they make the query.

Therefore, the optimal value of H, H^{opt}, is

$$H^{opt} = 1 - (1 - \frac{B}{N})^{M(k)} N \Big(\frac{1}{N} \sum_{i=1}^{N} \frac{\lambda_i}{\lambda}^{-\frac{1}{M(k)-1}}\Big)^{-[M(k)-1]}$$

We plot the optimal hit rates for different cache sizes for Zipf-distributed file request rates and $M(k) = 4k^2$ in Fig. 4. These results indicate that increasing the hop limit shows diminishing returns and, hence, the system designer should select a hop limit that is no more than the minimum desired to achieve the target hit rate. To understand if these optimal hit rates can be achieved, we plotted the optimal cache probabilities for one case, $N = 400, B = 20$, in Fig. 5. As we can see, as the hop-limit increases, the optimal cache probabilities begin to become more uniform (since enough nodes are being queried at high hop distances, it is sufficient to have cache probability below 0.1 for even the most popular file to have a very high probability of finding the file).

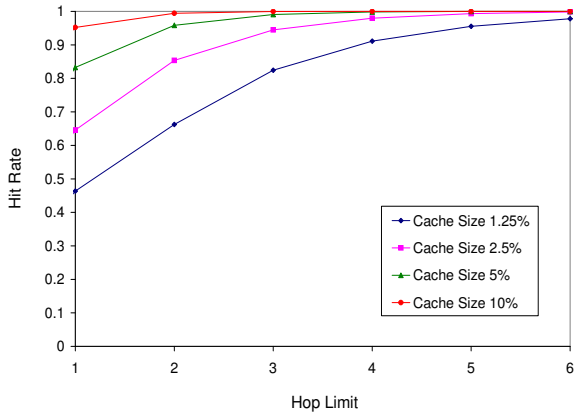


Fig. 4. **Distribution of Optimal Hit Rates with respect to Hop Count for varying Cache sizes with** $M(k) \sim 4k^2$

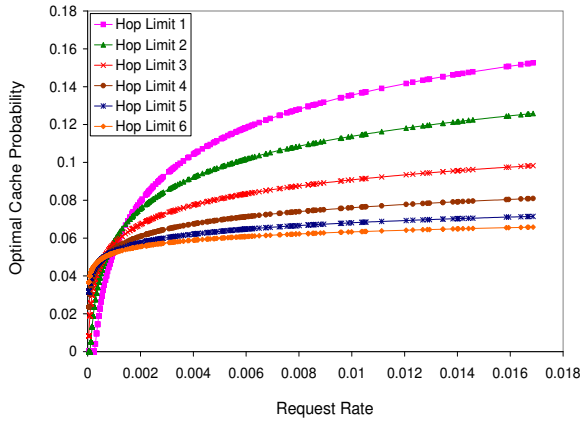


Fig. 5. **Distribution of Optimal Cache Probability with respect to Hop Count for varying request rates with** $M(k) \sim 4k^2$

While this optimization provides us with the best case hit rate performance, as we mentioned earlier, the number of replicas of each file are driven mainly by the file request patterns so it may not be possible to achieve the required cache probabilities needed for the optimal hit rates. Even more importantly, we must bear in mind that finding one source for the searched file (as measured by the hit rate) is not the end-goal. A near-uniform distribution of cache probabilities (as suggested by, say, a hop-limit of 6 in Fig. 5) implies that the sources that have the most popular file will have to serve far more requests than sources that have files that are, say, 10 times less popular as there are almost an equal number of sources of the two files. Such asymmetric distribution of download load is likely to result in queueing delays for the downloads for popular files. As shown in [13], optimizing download time performance requires a linearly proportional replica distribution. Further, in mobile wireless networks, downloading from a single source is unwise as the source peer can go out of range in a short time. Hit rate emphasizes finding one source of a file and gives no weight to finding multiple sources of a file and our design should not be solely guided by the hit rate performance. As [14] discusses, LRU gives a file replica distribution that is close-to-linearly proportional but populates fewer-than-linearly-proportional replicas of popular files. Thus, while LRU may be sub-optimal for download time performance, it is better than a linearly proportional replica distribution for hit rate. Since, LRU is also a popular cache management policy for many other reasons, we evaluate the hit rate performance with LRU next.

B. LRU Model

To determine the probability of finding the desired content in the selected hop limit, we need to find the probability of finding the file at any one node (since the request rates are uniform across nodes, the probability of finding a file is the same across the network). Since each node is same in all respects, an analytical model of the network of LRU-managed caches can be constructed with a single cache that, in addition to serving the local requests, also serves requests from remote nodes. We model this cache from the perspective of a particular file, say, file i - all requests for file i move the file to the top-most position in the storage; a request for any other files moves file i down to one lower position. Reference [3] presents an analytical framework for estimating the hit rate in stand-alone LRU-managed cache. By including the effect of remote requests, their model can be extended to model a network of LRU caches [7].

A critical component of this framework is $r(i, j, k)$, the rate at which file i is pushed down from position j to position $j + 1$ when the hop limit for scoped flooding is k hops. Let $p_{local}(i, j, k)$ be the probability of finding file i in top j positions in the cache when the hop limit is k hops. The probability of finding file i in local cache given a hop limit of k is then $p_{local}(i, B, k)$. $p_{local}(i, j, k)$ can be expressed in terms of $r(i, j, k)$ [3] by:

$$p_{local}(i, j, k) \approx \frac{r(i, j, k)}{\sum_{i=1}^{N} r(i, j, k)}$$

At steady-state, the push-down rate for file i from position j to $j+1$, $r(i,j,k)$, must equal the rate at which file i is brought into top j positions of the LRU stack (otherwise the probability of finding the file in these top j positions becomes unbounded). This conservation of flow principle helps us compute $r(i,j,k)$. File i is brought into top j positions under two conditions: (i) a local request for file i when file i is not in top j positions: the file may be brought to the top position from positions $j+1 \cdots B$ of the local cache if it is available there or it may be brought from a remote node (if a node within the search range has the file), this is $r_{local}(i,j,k)$; (ii) a remote request for file i: since the file i is not in top j positions, it must be in the remaining $j+1 \cdots B$ positions in the local cache for it to show up in top j positions on a remote request. Thus, we can write the following equations:

$$r_{local}(i,j,k) = \lambda_i[1 - p_{local}(i,j,k)]$$
$$[1 - (1 - p_{local}(i, B-j|j,k)(1 - p_{remote}(i,j,k)]$$

$$r_{remote}(i,j,k) = \lambda_i[1 - p_{local}(i,j,k)]$$
$$p_{local}(i, B-j|j,k)$$

where

$$p_{local}(i, B-j|j,k) = \frac{p_{local}(i,B,k) - p_{local}(i,j,k)}{1 - p_{local}(i,j,k)}$$

and

$$p_{remote}(i,j,k) = [1 - (1 - p_{local}(i,j,k))^{M(k)}]$$

A node sends out a file request only when it does not have the file. Thus, the rate at which the other $M(k)-1$ nodes send a file request for this file to the peer-to-peer network is $\lambda_i[1 - p_{local}(i,B,k)]$, where $p_{local}(i,B,k)$ is the probability that the file i is available at a node. The nodes that have file i in their cache satisfy these requests for file i sent to the peer-to-peer network. Assuming that the requests are uniformly distributed over the nodes that have the file, the request rate for file i served by a node that has file i on account of requests from other nodes equals $\frac{(M(k)-1)r_{remote}(i,j,k)S}{M(k)p_{local}(i,B,k)}$. Thus,

$$r(i,j,k) = r_{local}(i,j,k) + \frac{(M(k)-1)r_{remote}(i,j,k)S}{M(k)p_{local}(i,B,k)}$$

Starting with $p_{local}(i,1,1) = \lambda_i$ we can iteratively solve the above equations until the value of $p_{local}(i,B,k)$ converges. The complexity is $O(NB)$ and, in our computations, the value of p converged in only a few iterations.

Given $p_{local}(i,B,k)$, we can compute the hit rate for file i in the k-hop neighborhood as $P(i,B,k) = [1 - (1 - p_{local}(i,B,k))^{M(k)}]$ and the overall hit rate (across all searches) as:

$$P(B,k) = \sum_{i=0}^{N} \frac{\lambda_i}{\lambda}[1 - (1 - p_{local}(i,B,k))^{M(k)}]$$

Among the inputs to our model, the cache size B and the hop limit k are the design choices while λ_i, the file request rate distribution, and $M(k)$, the number of nodes in the k-hop neighborhood, are inputs that the system designer must provide for the specific application scenario being investigated.

We show the LRU performance for the dense, urban scenario and the sparse highway scenario in Figs. 6 and 7 respectively for Zipf-distributed file request rates. In Figures 6, 7, the cache ratio refers to the size of the individual node cache with respect to the total number of files in the network. So for example, a cache ratio of 0.1 means, an individual nodes' cache can store 10% of the total files in the network.

We find that with increasing hop count, the marginal gain in hit rates diminishes. This effect is even more pronounced as the cache ratio increases. Our analytical framework can be used to tune the query flood to achieve required levels of hit rates, and consequently the performance of AdTorrent by suitably adjusting the hop limit of the query flood. So, for example, if 80% hit rate was a satisfactory level of performance measure, our results suggest that a query hop limit of 4 will yield satisfactory performance in the dense-urban scenario irrespective of the cache size (as long it is above a certain threshold). Recall that our mobility model simulations in Section IV also suggested a hop limit of 4 to obtain a reasonable average connectivity duration which would facilitate robustness in protocol performance.

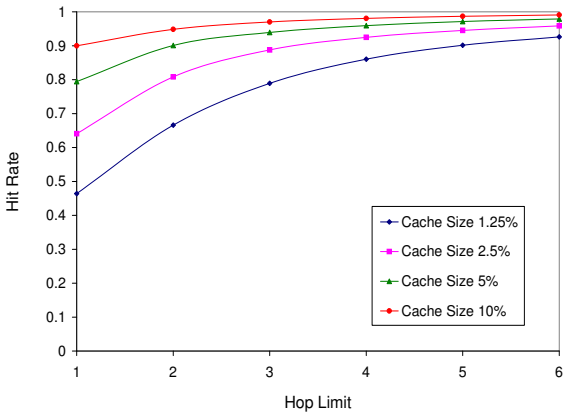


Fig. 6. **Hit Rate vs. Hop Count with LRU, for varying Cache sizes and** $M(k) \sim \alpha k^2$

C. Effect of Spreading Factor

As already discussed, downloading from multiple sources in mobile wireless networks is preferable. However, since parallel downloading affects cache probabilities, we wish to check the effect of increasing the spreading factor on the hit rates. In Fig. 8, we show the hit rates for different spreading factors. As we can see from the figure, the spreading factor has no effect on the hit rate so our choice of the spreading factor is not limited by hit rate considerations.

D. Hop Limit Selection

We can see from Figs. 6 and 7 that the cache size is an important factor in determining the hit rate and, thus in determining the appropriate hop limit. For example, if the

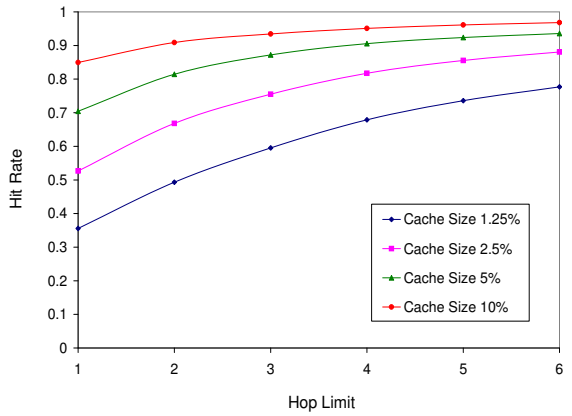


Fig. 7. **Hit Rate vs. Hop Count with LRU, for varying Cache sizes and sparse node growth** $M(k) \sim \alpha k^{1.4}$

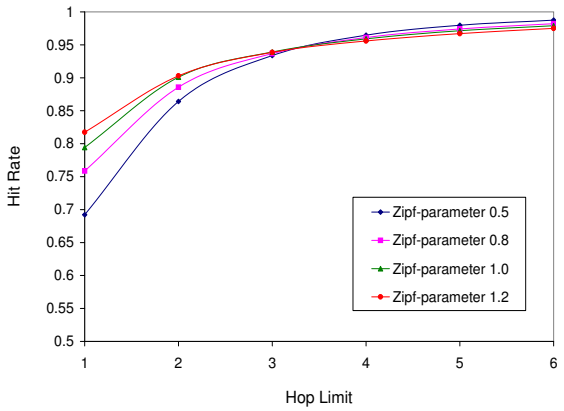


Fig. 9. **Effect of skew in request rates on the hit rate,** $M(k) \sim \alpha k^2$**, Cache Size 5%**

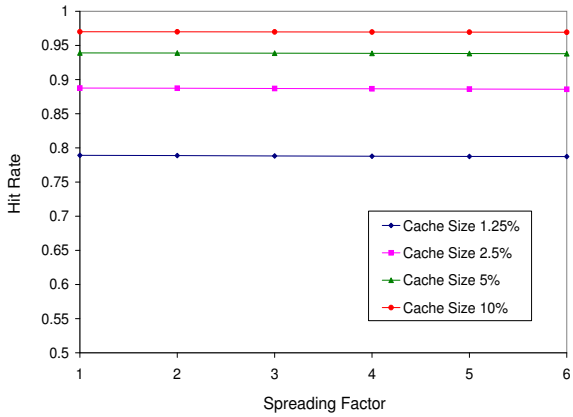


Fig. 8. **Effect of spreading factor on hit rate,** $M(k) \sim \alpha k^2$**, Hop Limit 3**

cache size is only 1.25% of the total number of files, we need information on about 60 nodes to achieve better than 75% hit rate and, as our content list aggregation gives us information on one extra hop, we need a hop limit of 6 in the sparse highway scenario (notice that, in the denser and faster-growing urban scenario, a hop limit of 3 would be sufficient to achieve this hit rate). In contrast, if the cache size was 5% of the total number of files, even with a hop limit of 2 in the sparse highway scenario (i.e. with querying only 11 peers per query) we get a hit rate of 80%.

E. Effect of Skew in File Request Rates

Skewed data access patterns improve the system performance in caching applications. To determine the sensitivity of hop limit selection on the skew in request rates, we computed the hit rate for different request rate distributions using our LRU model. The results are shown in Fig. 9. As expected, decrease in skew does decrease the overall hit rate but, unless the request rate has very little skew (e.g. has a zipf-parameter of 0.5), it may not be necessary to increase the hop limit or the cache size.

VII. Related Work

This section summarizes previous work related to cooperative data transfer protocols for the wired settings as well as vehicular environments. BitTorrent is a popular [2] file-sharing tool, accounting for a significant proportion of Internet traffic. There are two other peer-peer bulk transfer protocols namely, CarTorrent and Coopnet. CarTorrent [8] is a recent work that extends the BitTorrent protocol to the vehicular networks scenarios addressing issues such as intelligent peer and piece selection given the intermittent connectivity and limited bandwidth of the wireless medium.

Peer-to-peer networking in cooperative mobile environments has been proposed by several others. However, the constraint of limited buffers at the peers is discussed by very few others. [7] analyzed epidemic information dissemination to support web accesses with limited buffers per peer. Our analytical model to study the trade-off between the query hop limit and the overall hit rate is very similar to theirs. Since our problem setting is different from theirs, some of our assumptions are different. As discussed earlier, our model (as well as that of [7]) is an extension of the analytical model for a stand-alone LRU cache given in [3]. Limited buffers in case of peer-to-peer networks in wired scenario are also discussed in [14] which also uses a similar model for network of LRU caches but their end goal is different from ours.

VIII. Conclusion

In this paper we presented a novel application involving search and location aware content delivery(in our case advertisements/deals) to the nodes in a Vehicular Adhoc Network. We proposed a efficient keyword search on this content overlay. To aid system designers in selection of the design parameters in their implementation of AdTorrent application, we present a realistic mobility model for the urban, vehicular scenario and an analytical model of the epidemic query dissemination to evaluate the impact of the scope of the query dissemination on the hit rate. We derived the optimal hit rate as a function of the cache size and the hop limit and then developed a model for performance with LRU cache

management when swarming-based content delivery is used. System designers can use our analytical framework to estimate the required cache size and scope of the query dissemination based on user performance requirements. In our evaluation of some local scenarios, we found that a hop limit of 4 hops gives an adequate hit rate and that the incremental gain from increasing the scope of the query flood beyond 4 hops was minimal. These results are very encouraging in that they show the feasibility of AdTorrent deployment in urban scenarios.

REFERENCES

[1] B. Bloom, Space/time tradeoffs in hash coding with allowable errors. CACM, 13(7):422-426, 1970.

[2] B. Cohen, *Incentives Build Robustness in BitTorrent*, in IPTPS 2003.

[3] A. Dan and D. Towsley, *An Approximate Analysis of the LRU and FIFO Buffer Replacement Schemes*, in Proceedings of ACM SIGMETRICS 1990.

[4] *Dedicated Short Range Communication Architecture*, www.astm.org/SNEWS/MAY_2004/dsrc_may04.html

[5] Omprakash Gnawali, *A Keyword-Set Search System for Peer-to-Peer Networks*, Masters Thesis, Massachusetts Institute of Technology, 2002.

[6] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, *A group mobility model for ad hoc wireless networks*, in Proceedings of ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM), August 1999.

[7] C. Lindemann and O. P. Waldhorst, *Modeling Epidemic Information Dissemination on Mobile Devices with Finite Buffers*, in Proceedings of ACM SIGMETRICS 2005.

[8] A. Nandan, S. Das, G. Pau, M.Y. Sanadidi and M. Gerla, *Cooperative Downloading in Vehicular Wireless Ad Hoc Networks*, In Proceedings of Wireless On-Demand Networks and Services, St. Moritz, Switzerland, Jan 2005.

[9] *QualNet user manual* www.scalable-networks.com

[10] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, Handling Churn in a DHT, in Proceedings of USENIX 2005

[11] A. Saha and D. Johnson, *Modeling mobility for vehicular ad-hoc networks* in Proceedings of ACM VANET 2004

[12] K.Tang, M. Gerla and R. Bagrodia, *TCP Performance in Wireless Multi-hop networks*, in Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, 1999.

[13] S. Tewari and L. Kleinrock, *On Fairness, Optimal Download Performance and Proportional Replication in Peer-to-Peer Networks*, in Proceedings of IFIP Networking, May 2005.

[14] S. Tewari and L. Kleinrock, *Proportional Replication in Peer-to-Peer Network*, to appear in Proceedings of IEEE INFOCOM 2006.

[15] B. Zhou, K. Xu and M. Gerla, *Group and swarm mobility models for ad hoc network scenarios using virtual tracks*, in Proceedings of MILCOM 2004

Adaptive Retransmission Policy for Reliable Warning Diffusion in Vehicular Networks

Francesco Giudici, Elena Pagani, and Gian Paolo Rossi
Information and Communication Department, Università degli Studi di Milano, Italy
E-mail: {fgiudici,pagani,rossi}@dico.unimi.it

***Abstract*— Use of wireless technologies is becoming pervasive in everyday life. Recently, research began analyzing their use on board of vehicles for several kinds of applications, ranging from traffic safety to fleet management and cooperative work, to entertainment and Internet browsing. In this work, we focus on safety applications and in particular on the approach proposed in the framework of the PATH project [1] for reliable diffusion of warnings to advertise problems in vehicular traffic. The approach in [1] is based on *static* parameters describing the environment. Unfortunately, in real environments those parameters may dynamically change over time. In this work we present performance measurements obtained by varying the parameters, to evaluate how the performance of the approach depends on environmental conditions.**

I. Introduction

Vehicles equipped with wireless network interface cards (WNICs) start to be available. This equipment can be used for several applications, ranging from fleet management and cooperative workgroup to entertainment and Internet navigation. In this work we focus on the problem of *vehicular safety*. Wireless networking can be exploited to provide communication among vehicles, in order to notify the occurrence of problems – e.g., accidents, icy street, obstacles on the road – to other oncoming vehicles. *Warnings* are addressed to all vehicles approaching the place where the problem occurred, so as to allow drivers to perform the appropriate actions. Warning traffic has service requirements that must be guaranteed by the system. *Low latency* is needed to guarantee that the warning can be detected by a driver so that s/he has sufficient time to properly react to the event notified. *High reliability* is needed to guarantee that all interested vehicles actually receive a warning. On the other hand, wireless links have some unfavorable characteristics, such as long latency for channel set-up and low reliability. The former is due to the time needed to two devices to synchronize and agree about the policy for channel access (such as the used code or frequency hopping pattern). Some solutions have already been proposed to exploit wireless technologies to supply vehicular safety, some of which supported by car producers and government institutions. In this work, we focus on the *static* approach proposed by the PATH project [1] for reliable diffusion of warnings, with the aim of both understanding how environmental characteristics impact on the obtainable reliability, and devising mechanisms to dynamically adapt the approach in order to optimize performance according to changes in those characteristics. The California PATH Project involves among other things, researches on an infrastructure to boost vehicular safety. The proposed solution aims at achieving reliable dissemination of warnings through *repetitions*, i.e., multiple retransmissions of a warning so as to overcome channel failures and collisions with other messages. PATH seems the most promising approach, and it is under study for adoption on U.S. highways. However, it assumes that an optimal number of repetitions exists for a certain scenario. Unfortunately, the vehicular environment is highly dynamic. The density of vehicles in a certain area, the number of concurrent warning sources and the vehicle speed vary over time, and so should do the number of repetitions. In this work, we analyze by simulations how the number of repetitions needed to achieve reliability varies depending on the characteristics of both the vehicular and the data traffic; an alternative adaptive policy is discussed. A Vehicular Collision Warning Communication (VCWC) [3] has been proposed, focusing primarily on achieving a low transmission latency. VCWC does not take reliability aspects into considerations. Both the above proposals rely on the DSRC (Dedicated Short Range Communications) multi-channel architecture [4]. DSRC has been explicitly designed for use in vehicular systems. DSRC proposes communication services for both private applications and public safety, with the possibility of using high power transmission when latency is important. DSRC is now in the process of standardization by IEEE as the WAVE (Wireless Access in Vehicular Environments) project; it is also known as the ISO CALM (Communications Air Interface Long and Medium range) standard [5]. The European Project CarTALK/Fleetnet [6], [2] uses UTRA-TDD (UMTS Terrestrial Radio Access with Time Division Duplexing) as the channel architecture, thus adopting a frequency range requiring licensing.

II. System Model

In this work we consider a system composed by vehicles equipped with wireless network interface cards (WNICs). Vehicles have a GPS system. We focus on vehicle-to-vehicle communication; no roadside communication infrastructure is needed. A vehicle can notify road hazards to all oncoming vehicles; communication is broadcast and addressed to one-hop neighbors only. Warnings may contain information about the zone affected by the notified problem. The wireless technology is based on the 802.11 standard [7]. In particular, the channel structure is determined by the DSRC proposal [4]. Vehicles are equipped with *On-Board Units* (OBUs)

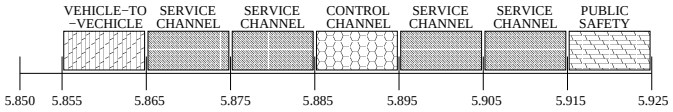


Fig. 1. Layout of the DSRC channel

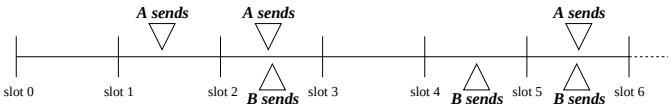


Fig. 2. Example of PATH protocol execution

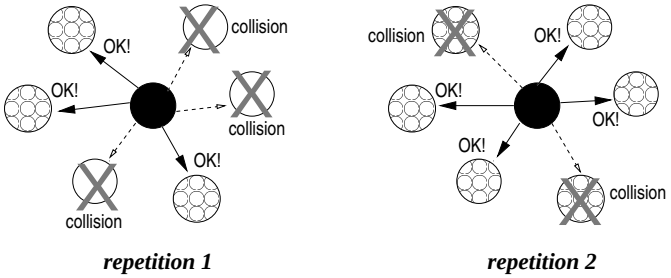


Fig. 3. Example of successful execution of PATH protocol

that support communications among vehicles. DSRC uses the transmission range 5.850 to 5.925 GHz. The transmission range is divided into 7 channels of 10 MHz each (fig.1); the data rate supported is up to 27 Mbps. MAC and physical layers are provided by the IEEE 802.11p proposal [8], [9]. DSRC has an average communication range of around 300 mt., and up to 1000 mt. Channel access is performed through CSMA. Channels have different aims: four of them are *Service* channels that can be used mainly for common data and private applications, but also for public safety. All Service channels are accessed in a shared way by all vehicles. The *Control* channel is mainly used to exchange control information needed to synchronize vehicles for access to the other channels and to announce the correspondence among applications and Service channels. It is also used to exchange high priority messages for vehicular safety. Time to access the Control channel must not be greater than 100 msec.; the channel must not fail in case of congestion. A device must listen to the Control channel for intervals of at least 200 msec., and it cannot be off the Control channel for more than 50 msec. A *vehicle-to-vehicle communication* channel exists, devoted for instance to publish information about the mobility pattern of a vehicle, in order to forecast the possibility of accidents and forewarn drivers. The last channel is dedicated to the exchange of warnings for *public safety*. All channels are used for several types of traffic and can be accessed by multiple vehicles concurrently; hence, collisions are possible. In this work we assume that all warnings notifying a problem in vehicular traffic are sent on the Control channel [10]. We analyze the problems involved with performing retransmissions on that channel in order to guarantee high reliability while at the same time avoiding channel congestion.

III. California PATH Project

In the framework of the California PATH project, six protocols have been proposed [11] to diffuse warnings for vehicular safety within a bounded time, while guaranteeing high reliability. Warning messages must be reliably received by all source's neighbors with a low latency. A warning has associated a *packet lifetime* τ within which it must be reliably received by all neighbors before becoming useless because too late to allow drivers to appropriately react: it is an upper bound on the transmission latency. Due to the broadcast diffusion of warnings, acknowledgments cannot be used to control reliability, to avoid ack implosion at the sender; for the same reason, the RTS/CTS mechanism cannot be used. The protocols proposed consists in considering the packet lifetime interval as divided into n *slots* such that $n = \lfloor \tau / T_x \rfloor$ where T_x is the transmission time of a warning, depending on the packet size and the channel bandwidth. Hence, a packet can be sent in a slot. The protocols presented in [11] differentiate in three respects:

- nodes can perform carrier sensing before accessing one of the chosen slots, or not. In the former case, if the channel is busy, the repetition scheduled for that slot is dropped, thus actually decreasing the number of repetitions performed;
- nodes are synchronized on slot beginning, or slots are locally determined according to the instant a warning is generated;
- when a node has a warning to send, either it a priori randomly chooses K slots among the n and tries to send the packet in those slots, or in each of the available slots transmits a warning with a uniformly distributed probability of K/n.

A warning has been reliably delivered when each node in the communication range of the source has received it at least once. It is worth to notice that, because of broadcast transmissions, a node can receive duplicates. The protocol fails if one or more source's neighbors exist that do not receive any warning. As an example, in fig.2, two sources are sending their warnings, and they performs 3 repetitions. They send their warnings in the slots chosen a priori and they collide in slots 2 and 5, while transmissions in slot 1 for source A and in slot 4 for source B are successful. Let us notice that, because of the hidden station problem, not necessarily a transmission is successful for *all* neighbors. In each repetition a source could reach only a subset of its neighbors. In fig.3, a situation is shown in which none of two repetitions is successful, but they together reach all destinations. What matters here is that throughout the K repetitions all neighbors have been reached at least once. According to simulation results discussed in [11], best performance has been achieved with slots for repetitions randomly chosen a priori, asynchronous nodes, and nodes performing carrier sensing before sending a packet in a slot (Asynchronous Fixed Repetition with Carrier Sensing, AFR-CS protocol). In fig.4, pseudo-code for AFR-CS is provided.

```
when (a warning must be sent) do
    for (i = 1 to K) slot[i] ← randomly chosen slot;
    for (i = 1 to K)
        when (current slot = slot[i]) do
            carrier sensing;
            if (slot free) then send i-th repetition;
        od
    end for
od
```

Fig. 4. Pseudo-code of AFR-CS

In [11], an analytical evaluation has been performed according to which the optimal number of repetitions is 7, under the hypothesis that warning generation follows a Poisson distribution and for a specific scenario with communication range of 80 mt., average distance among vehicles 30 mt., 4 lanes, and 75 interferers around each receiver. It is extremely important to carefully estimate an appropriate value for the number of repetitions. Reliability must be obtained, but without risk of congesting the Control channel, which *must* remain available for its other usages. However, analytical evaluation of K does not seem appropriate: in the considered *highly dynamic* environment it is impossible to characterize an average situation so as to optimize K. The analysis bases on assumptions that are not necessarily valid in a real environment, such as poissonian generation of warnings or estimation of the number of interferers. The number of interferers is not the same for all recipients. As a consequence, this approach can be used in a real environment only by configuring parameters basing on an average situation, which could be far from the actual situation, thus leading either to low reliability – for too low K – or to both congested network and low reliability – for too high K. As an alternative, a vehicle should consider the current state of the vehicular traffic, and dynamically adapt the number of retransmissions needed to disseminate its own warnings basing on local observations about the number of neighbors and the load of warning traffic in its surroundings in the recent past.

IV. ENHANCING PATH

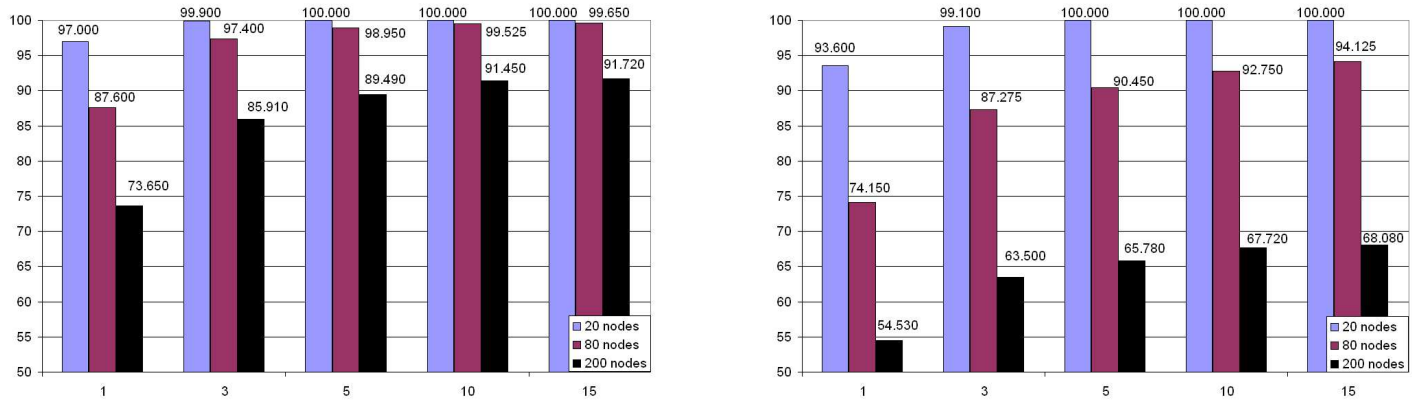
We performed simulations of PATH using the NS-2 package to highlight correlations among the vehicular and data traffic conditions and the achieved reliability. The parameters used to evaluate PATH performance are shown in Table I, and they are the same adopted in [12]. The aim of the measures is to evaluate an upper bound on the retransmissions needed to achieve reliability by stressing the system. In simulations, the message generation interval has been set equal to the packet lifetime so as to guarantee that each node is always sending a warning. The channel bandwidth has been set to 18 Mbps, in accordance with the considerations reported in [12]. In [5] a data rate of 6 Mbps is assigned to the Control channel, while the other channels have a data rate of 27 Mbps; we also performed measures with 6 Mbps rate. Devices are distributed over a small area so that are all in communication range. Packet size allows to communicate coordinates – according to a GPS system – indicating where a problem occurred. From packet size and 18 Mbps bandwidth, a slot time of 147 μsec. is obtained, slightly greater than the packet transmission time. As a consequence, the number of slots is $n = (\tau/$ slot time)$= 681$. The communication range is in line with the DSRC characteristics. The message range, that is, the distance from the source at which the message should be propagated, equals the communication range, thus enforcing one-hop diffusion. Mobility impacts on the definition of reliability, because vehicles near the warning source can move out of communication range before receiving the packet, and vehicles can enter the source communication range within the packet lifetime. To accurately measure the reliability degree without having to deal with mobility issues, vehicles do not move in our simulations. Measures have been performed with 20, 80 and 200 nodes in the network, for variable number of repetitions.

TABLE I
PARAMETERS FOR PATH PERFORMANCE EVALUATION

Packet lifetime (τ)	100 msec.
Message generation interval	100 msec.
Packet size	250 Bytes
Control channel bandwidth	18 Mbps
Communication range	250 mt.
Message range	250 mt.
Mean distance among neighbors	$\leq$ 250 mt.
Slot time	147 μsec.

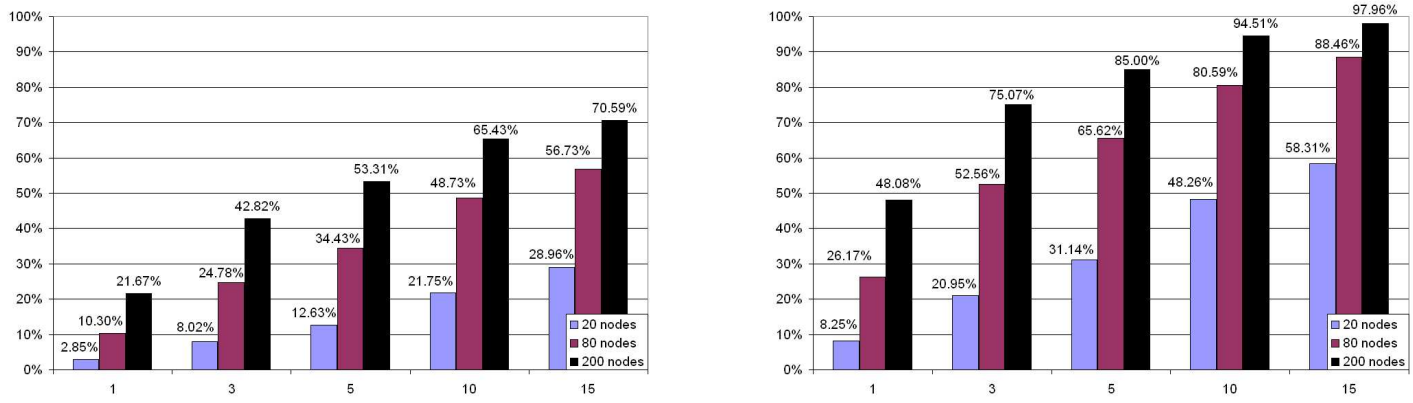
A. Performance Analysis and Optimization

For both values of control channel bandwidth, simulation conditions exist in which 100% reliability cannot be achieved (fig.5). For high number of nodes, increasing the number of repetitions is not effective when the channel tends to congest, and the achieved reliability tends to stabilize. Channel congestion (fig.6) increases almost to saturation. For larger (18 Mbps) bandwidth and 15 repetitions still 1/3 of the channel is unused although reliability has already stabilized. This is due to a greater probability that nodes choose the same slots to perform repetitions. Indeed, the probability of a slot to be chosen by a certain node to send a repetition is the ratio (number of repetitions / n), which for 15 repetitions amounts to 0.02 for 18 Mbps and 0.06 for 6 Mbps. Hence, The probability for a slot of being not used by any of 200 nodes is $0.98^{200} \simeq 0.018$ in the former case, while $0.94^{200} \simeq$ 4E-6 in the latter. The carrier sensing mechanism helps in avoiding collisions (fig.7), but it confirms channel congestion. For high number of both nodes and repetitions, often a certain slot chosen a-priori cannot be used for sending a repetition because already in use. Behavior for 6 Mbps bandwidth is similar; for 200 nodes and 15 repetitions the probability of finding a slot already in use increases up to 92.56%. On the other hand, also under critical conditions the probability of collisions is negligible (fig.$7(b)$). It is worth to notice that, in case nodes



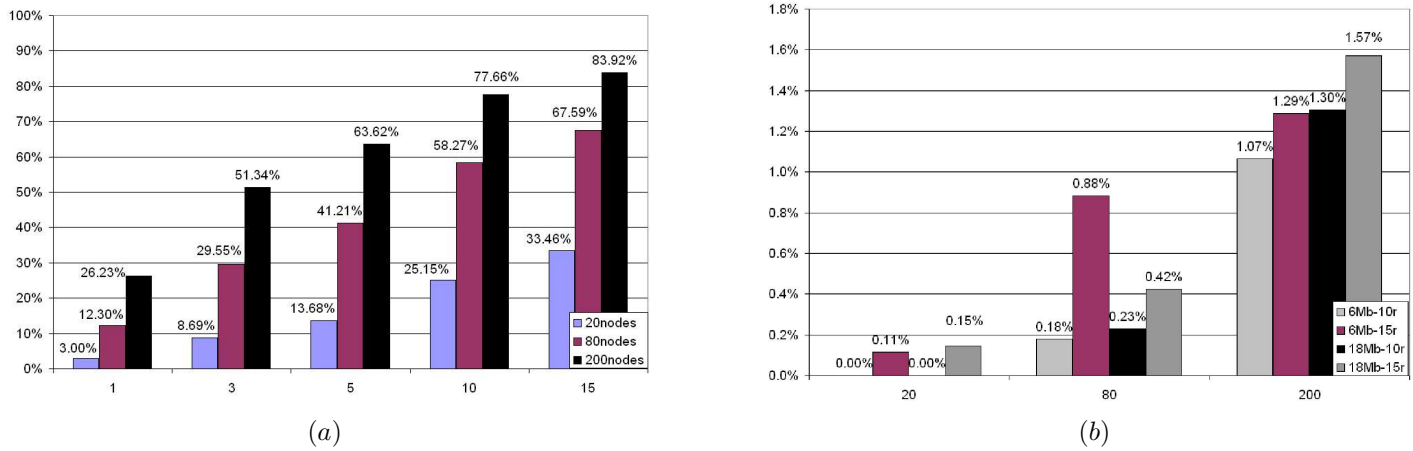
(a) (b)

Fig. 5. Percentage of warnings reliably delivered with respect to number of repetitions issued by each node for (a) 18 Mbps or (b) 6 Mbps of channel bandwidth



(a) (b)

Fig. 6. Percentage of slots used for (a) 18 Mbps or (b) 6 Mbps of channel bandwidth, with respect to number of repetitions issued by each node



(a) (b)

Fig. 7. (a) Percentage of slots found busy with 18 Mbps with respect to number of repetitions issued by each node. (b) Percentage of collided packets with respect to number of nodes

are not all in range, the *"hidden station"* phenomenon would increase collisions, which could be much more disruptive for reliability than repetitions suppressed because of busy channel. We analyzed the contribution of each repetition to the globally obtained reliability. In fig.8, the percentage of times in which reliable delivery has been achieved in the i-th repetition is reported. The percentage has been evaluated over a number of warnings equal to ($50 *$ number of nodes). For a few nodes, there is greater probability that they choose different slots to perform repetitions. Hence, all destinations are reached with a low number of retransmissions. By contrast, the greater the number of nodes, the greater the number of retries before achieving reliability. Indeed, nodes contend for using the same slots: with 18 Mbps bandwidth, 200 nodes and 15 repetitions for each warning, the number of slots needed to accommodate repetitions of all nodes is $200 \times 15 = 3000$ while only 681 slots are available in a packet lifetime. A slot could be thus chosen by 4-5 sources on average. One of them succeeds in accessing the channel, while the others omit to perform a repetition and wait for the next slot chosen. As a consequence, the average number of repetitions needed to achieve reliable delivery increases for increasing number of nodes. Moreover, increasing the number of repetitions, the probability of succeeding with the first repetition decreases; but on the other hand increases the probability of success in successive repetitions, thus yielding a better global reliability than with only a few repetitions.

Several considerations can be inferred from the results presented above. First of all, a statically determined number of repetitions (7 according to the analysis performed in [1]) is not always adequate. For instance, with low number of nodes less repetitions (3 -5) are enough to reliably diffuse warnings, without at the same time congesting the Control channel. The most appropriate decision for a node seems to be performing enough repetitions to reach a stable reliability without high congestion. Further dissemination of information about the traffic event signaled by the node could be obtained by warnings generated by other nodes detecting the same event, or by warning generated as a consequence of the original warning. On the other hand, congestion on the control channel *must* not occur to avoid making non-accessible the other channels. Indeed, the simplest solution to guarantee high reliability in any condition would be to exploit carrier sensing: a node continuously senses the channel and sends a repetition in each slot it finds unused, possibly till the desired number of repetitions has been reached. But this approach is absolutely not suitable in order to guarantee proper work of other channels. Each node should monitor the number of other nodes in its neighborhood that are generating warning, and the traffic load, and compute its number of repetitions according to those parameters. The number of repetitions should be dynamically adapted according to variations in the number of neighbors. The most promising solution, and the one we are going to evaluate with further simulations, allowing to reduce congestion and contention on slot usage, consists in having a node that refrains from performing all the repetitions initially scheduled in the event it senses the channel free for the first few (3-5) repetitions, thus letting the channel usable by other nodes. On the other hand, a node that senses the channel busy could re-schedule the suppressed repetition in one of the successive slots in order not to decrease its probability of success for inability in using the channel. Although re-scheduling could be computationally heavy. A better comprehension of the mechanisms coming into play in the described system could be obtained by devising a statistical model of the behavior of nodes and performing an analytical evaluation. Yet, many phenomena may impact on both achieved reliability and channel usage, which cannot be easily modeled analytically, nor reproduced in simulations. They are discussed in the next session.

B. Behavior in Real Environments

Simulations show that the number of retransmissions needed to achieve reliability depends on the load offered to the network and on the density of devices. Because of mobility, each node may observe dynamic changes of these indexes as a consequence of its own movements and the movements of the devices in its communication range. As an explicit requirement of the DSRC architecture is that the Control channel is resilient to congestion, the number of retransmissions must be the lower bound needed to achieve reliable warning delivery. A dynamic policy – able to adapt to the current network state – is preferable to a static one both to supply reliability guarantees and to avoid congestion. As a matter of fact, in real environments many other situations occur, which are very difficult to reproduce with simulations. A warning reporting a traffic problem may be – almost simultaneously – generated by all vehicles near the position of occurrence and detecting the problem. One one hand, these *duplicate* warnings compete to use the channel, thus making more difficult guaranteeing a reliable delivery of all of them. On the other hand, as they signal the same problem, it is enough that a vehicle receives at least one of those warnings from one of the advertisers. Hence, *multi-path* propagation helps achieving reliability. It is difficult to evaluate the extent to which these competing effects impact on vehicles (drivers) behavior. A warning can trigger the generation of *cascading* warnings. In case a driver suddenly brakes, his/her vehicle V sends a warning to oncoming vehicles, let us say W and Z. Those vehicles in turn are forced to brake or slow down, generating on their behalf other warnings. This chain of events propagates the notification of a traffic problem over multiple hops. But vehicles in the communication range of V, W and Z receive different warnings concerning the same problem. This phenomenon contributes in increasing reliability. It is worth to notice that in all our simulations only warning traffic has been generated. In fact, in a real DSRC environment the Control channel is also used by other data traffic,[1] and those messages are not subject to retransmissions as they do not have reliability requirements. This has a twofold consequence: (i)

[1]E.g., announcements of services available on the other channels.

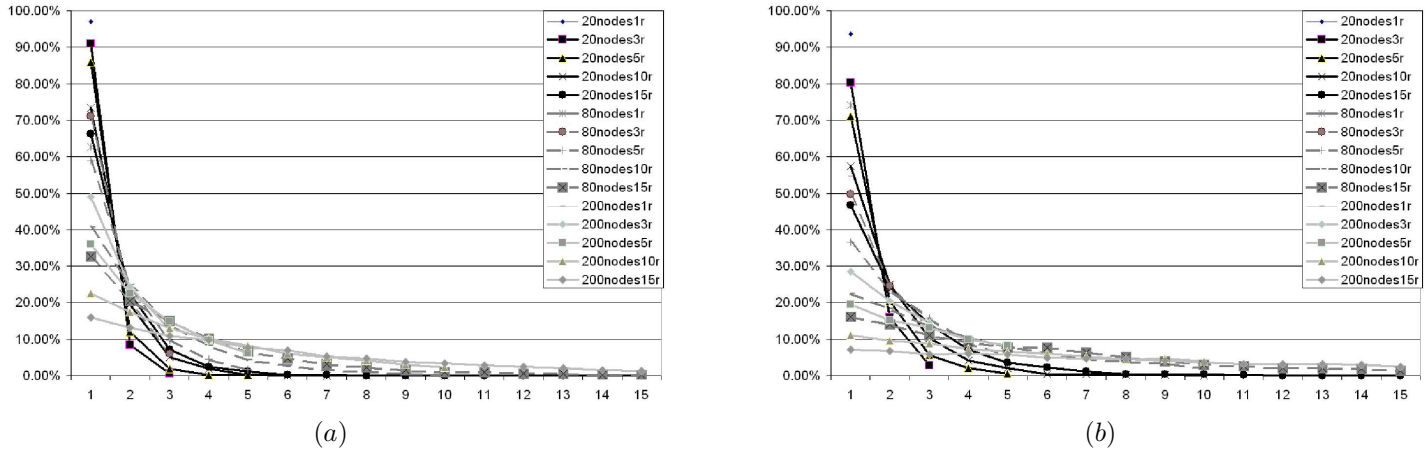


Fig. 8. Percentage of warnings that have been reliably delivered at the i-th repetition, for (a) 18 Mbps or (b) 6 Mbps of channel bandwidth

concurrency in medium access should be lower than that we reproduced, also in conditions of high vehicular density; (ii) data traffic with no reliability requirements risks to be pushed out of the network because of the aggressiveness of warning traffic. As far as the latter issue is concerned, as data traffic sent over the Control channel is needed to synchronize accesses to Service channels, if nodes cannot access the media then the whole system is disrupted. A solution could be to equip vehicles with two WNICs – according to WAVE specification. One antenna is devoted to safety applications while the other one is used for all other applications. In this case, concurrency among warnings could be accurately modeled by the presented simulations.

V. Conclusions and Future Works

In this work, an approach is analyzed for warning dissemination in vehicular networks, with the purpose of deriving indications to make it adaptive. Measurements provide several ideas about how to dynamically change node behavior according to current vehicular traffic and network conditions, and what parameters to consider for this purpose. These ideas must be validated by further simulations. Moreover, other future developments can be imagined. A warning is addressed to one-hop neighbors of the source. Depending on the vehicle speed, this could be not enough, for instance if the speed is so high that the route covered by a vehicle before arriving to the place a problem occurred – or needed to a driver to brake before arriving there – is larger than the communication range. In these cases multi-hop propagation is needed. In the discussed simulations, warnings are unrelated one to another. A more careful analysis could be performed to highlight whether correlated, cascading warnings are effective to propagate a warning over multiple hops in acceptable time. An alternative approach we are exploring is to set-up ad hoc *safety networks* dedicated to the exchange of warnings, so that a vehicle always belong to a safety network and is able to receive warnings of interest. Such an approach must cope with the delays involved in creating, joining and merging ad hoc networks, and it seems to require amendments to the 802.11 standard. Further simulations must be performed to evaluate the mutual impact of warning traffic and all other traffic. On one hand, concurrency among several traffic flows would make more difficult to provide reliability guarantees. On the other hand, it is interesting to measure how repetitions for warning messages affect normal traffic, in order to ensure a fair bandwidth usage among flows, compatibly with respective service requirements. Moreover, the effects of mobility could be analyzed for different vehicle speeds, once an appropriate reliability definition is characterized for the case of changes of the destination group.

References

[1] California PATH Project, *Partners for Advanced Transit and Highways (PATH)*. `http://www.path.berkeley.edu/`.

[2] W.J. Franz, H. Hartenstein, B. Bochow, *Internet on the Road via Inter-Vehicle Communications*. Proc. Workshop der Informatik 2001: "Mobile Communications over Wireless LAN – Research and Applications", Sep. 2001, `http://www.et2.tu-harburg.de/fleetnet/english/documents.html`.

[3] X. Yang, J. Liu, F. Zhao, N.H. Vaidya, *A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning*. Proc. 1st IEEE Annual Intl. Conf. on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), 2004, pp. 114-123.

[4] DSRC writing group, *Dedicated Short Range Communications (DSRC) – Tutorial*. `http://www.leearmstrong.com/DSRC/DSRCHomeset.htm`.

[5] IEEE, *Consolidated Report on the Requirements for Public Safety Security in WAVE Systems*. Draft 0.8, IEEE, June 15 2004.

[6] CarTALK Consortium, *CartTalk 2000 Project*. `http://www.cartalk2000.net`.

[7] IEEE Standards Association, *IEEE 802.11 Wireless Local Area Networks*. `http://grouper.ieee.org/groups/802/11/index.html`.

[8] IEEE Vehicular Technology Society, *5.9 GHz Dedicated Short Range Communication (DSRC) – Overview*. `http://grouper.ieee.org/groups/scc32/dsrc/index.html`.

[9] B. Cash, *WAVE Background Information – Wireless Access in Vehicular Environments for the 5.9 GHz band*. doc.: IEEE 802.11- 04/ 0121r0, Jan. 2004.

[10] R. Sengupta, Q. Xu, *DSRC for Safety Systems*. Intellimotion – Research Updates in Intelligent Transportation Systems, Vol. 10, n. 4, 2004, pp.2.

[11] Q. Xu, T. Mak, J. Ko, R. Sengupta, *MAC Protocol Design for Vehicle Safety Communications in Dedicated Short Range Communications Spectrum*. Proc. IEEE ITSC 2004, `http://path.Berkeley.edu/dsrc`.

[12] Q. Xu, T. Mak, J. Ko, R. Sengupta, *Vehicle-to-Vehicle Safety Messaging in DSRC*. Proc. 1st ACM Workshop on Vehicular Ad Hoc Networks (VANET'04), 2004, pp. 19-28.